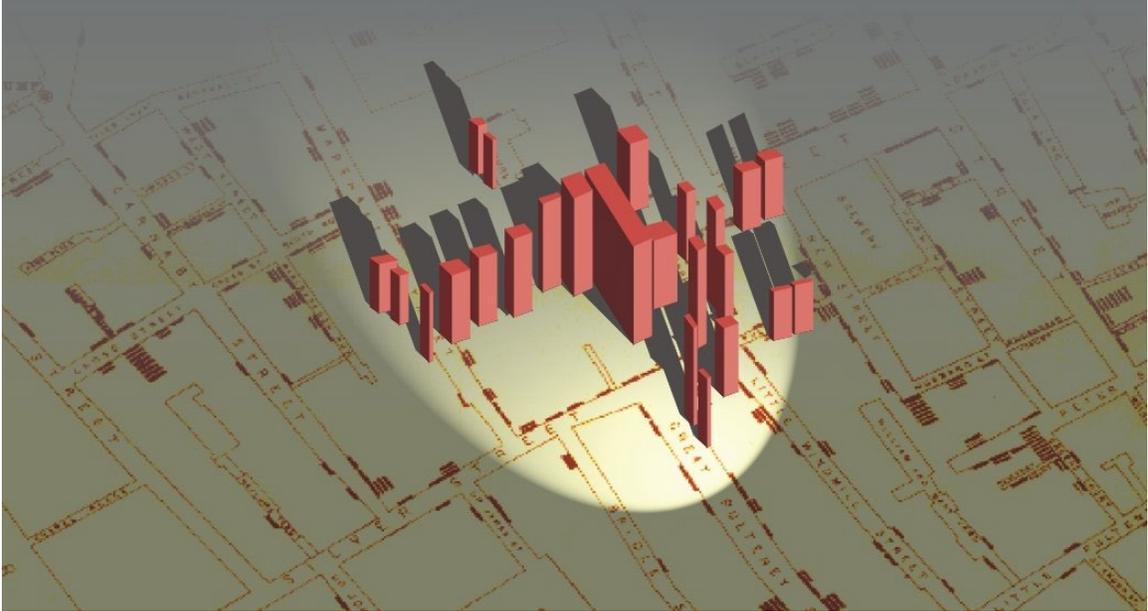


Epilnfo™

2000

Manual



Epi Info™ 2000

**A Database, and Statistics Program
for Public Health Professionals
Using Windows® 95, 98, NT, and 2000 Computers**

**Program design by Andrew G. Dean, MD, MPH,
and Thomas G. Arner, PhD (*Epi Map*)**

**Manual by Andrew G. Dean, Juan Carlos Zubieta, MD,
MPH and Kevin M. Sullivan, PhD, MPH, MHA, and
Cecile Delhumeau**

**Programming by Godha Sunki, MS, Sireesha Sangam,
Thomas G. Arner, Roger Friedman, Matthew Lantinga,
Andrew G. Dean, Saliil Diskalkar, and Donald C. Smith**

**Testing by Juan Carlos Zubieta, Catherine Schenck-
Yglesias, MHS, Cassandra Brown, and Natalie Huet**

**Tutorial exercises by Juan Carlos Zubieta, Consuelo M.
Beck-Sagué, MD, and G. Allen Tindol, MD**

**Division of Public Health Surveillance and Informatics
Epidemiology Program Office, MS K74
Centers for Disease Control and Prevention (CDC)
Atlanta, Georgia 30341-3717**

Previous versions produced in collaboration with the World Health Organization (WHO), Geneva, Switzerland, by Andrew G. Dean, Jeffrey A. Dean, Denis Coulombier, Anthony H. Burton, Karl A. Brendel, Donald C. Smith, Richard C. Dicker, Kevin M. Sullivan, Thomas G. Arner, and Robert F. Fagan.

This manual and the programs are in the public domain and may be freely copied, translated, and distributed. They are available on the Internet at www.cdc.gov/epiinfo. Small corrections were made on November 30, 2000.

Suggested citation: Dean AG, Arner TG, Sangam S, Sunki GG, Friedman R, Lantinga M, Zubieta JC, Sullivan KM, Smith DC. Epi Info 2000, a database and statistics program for public health professionals for use on Windows 95, 98, NT, and 2000 computers. Centers for Disease Control and Prevention, Atlanta, Georgia, USA, 2000.

**Epi Info Hotline for Technical Assistance
epiinfo@cdc.gov
(770) 488-8440
FAX (770) 488-8456**

Epi Info 2000

Acknowledgements

Environmental Systems Research Institute, Inc. (ESRI), of Redlands, California, the publishers of ArcView and ArcInfo, provided a special waiver for royalty-free distribution of the MapObjects mapping engine contained in *Epi Map 2000*, on condition that *Epi Map 2000* is distributed only with Epi Info 2000. Robert K. Jung, ARJ Software, Norwood, Massachusetts, gave permission for use of ARJ compression programs used in distributing translations. Dr. David Martin, Brookline, Massachusetts, and A. Ray Simons, Atlanta, Georgia, provided Turbo Pascal procedures for exact confidence limits. Mr. Jan Verhoeven (<http://jans.hypermart.net>) gave permission to distribute his freeware program, Andante4, with Epi Info on CD-ROM. The programs for logistic regression and Kaplan-Meier survival analysis are based on the MULTLR and KMSURV programs of Eduardo Franco, PhD, and Nelson Campos-Filho, with their kind permission and collaboration. We thank the many beta testers who provided feedback and suggestions.

Notes

These programs are provided in the public domain to promote public health. Please give copies of the programs and the manual to friends and colleagues. The programs may be freely translated, copied, distributed, or even sold without restriction except as noted below. No warranty is made or implied for use of the software for any particular purpose.

“Epi Info” is a trademark of the Centers for Disease Control and Prevention (CDC). Please observe the following requests:

- When distributing more than 10 copies of Epi Info, please report to the Epi Info Help Line the number of copies distributed, the countries to or in which they were distributed, and whether the distribution was by Internet or CD-ROM.
- The programs can be translated and the examples altered for regional use, but the programs must be distributed in essentially the form supplied by CDC. *Epi Map 2000* cannot be distributed separately from Epi Info 2000.

Epi Info is written in Visual Basic, Version 6, from Microsoft, Inc. The manual was written and indexed in Microsoft Word. The cover art is by Pete Seidel, Public Health Practice Program Office, CDC, and Dr. John Snow, London. The master disks were tested for computer viruses with the McAfee virus-detection software (McAfee Associates, (408) 988-3832).

Microsoft, Windows, Word, and Visual Basic are registered trademarks of Microsoft Corporation and ArcView and ArcInfo of Environmental Systems Research Institute, Inc. (ESRI). Trade names are used for identification only or for examples; no endorsement of particular products is intended or implied. The use of trade names or trademarks in this manual does not imply that such names, as understood by the Trade Marks and Merchandise Marks Act, may be used freely by anyone.

Epi Info 2000

Technical Support

For new versions of the software and answers to commonly asked questions, please visit the Epi Info web site at www.cdc.gov/epiinfo/. Technical assistance is provided by e-mail, telephone, or FAX. Information for obtaining Epi Info technical assistance is given on the title page.

The Epi Info Worldwide Discussion Group (LISTSERV) provides a forum for user questions and answers. Join the group by sending e-mail to listserv@listserv.cdc.gov, with the words *Subscribe epi-info* in the body of the text. A message containing further instructions will be sent in response.

Please send **comments and suggestions** for future versions to:

Andrew G. Dean, M.D., M.P.H.
Epidemiology Program Office, Mailstop K74
Centers for Disease Control and Prevention
Atlanta, GA 30341-3717
agd1@cdc.gov
Telephone (770) 488-8432
Home Telephone (770) 458-2271
(Emergencies only please)

Epi Info 2000

Table of Contents

OVERVIEW OF EPI INFO 2000	7
INSTALLATION.....	11
WHAT'S NEW?	15
THE EPI INFO 2000 PROGRAMS	21
RAPID TOUR.....	29
GUIDED TOUR.....	31
DATA MANAGEMENT	49
HOW TO.....	61
SAMPLES.....	73
NUTSTAT	75
COMMANDS.....	89
FUNCTIONS AND OPERATORS	157
INTERNAL DETAILS	181
LIMITATIONS AND PROBLEM RESOLUTION	193
TRANSLATION	197
STATISTICS.....	203
LOGISTIC REGRESSION – THE MVAWIN PROGRAM.....	257
KAPLAN-MEIER SURVIVAL ANALYSIS.....	273
APPENDIX: RESOURCES FOR CREATING PUBLIC HEALTH MAPS	283
INDEX	291

Epi Info 2000

Overview of Epi Info 2000



What Is Epi Info 2000?

Epi Info is a series of programs for Microsoft Windows 95, 98, NT, and 2000 for use by public health professionals in conducting outbreak investigations, managing databases for public health surveillance and other tasks, and general database and statistics applications. With Epi Info and a personal computer, physicians, epidemiologists, and other public health and medical workers can rapidly develop a questionnaire or form, customize the data entry process, and enter and analyze data.

Epidemiologic statistics, graphs, and tables are produced with simple commands like READ, FREQ, LIST, TABLES, and GRAPH. A component called *Epi Map* displays geographic maps with data from Epi Info. Epi Info is in the public domain and can be downloaded from the Internet. CD-ROM copies and printed manuals are expected to be available from private vendors.

The first version of Epi Info was released in 1985. A study in 1997 documented 145,000 copies of the DOS versions of Epi Info and *Epi Map* in 117 countries. The DOS manual and/or programs have been translated into 13 non-English languages.

Epi Info 2000 is an entirely new series of programs for Microsoft Windows 95, 98, NT, and 2000, written in Visual Basic, Version 6. It uses the Microsoft Access file format as a gateway to industry database standards. Although Epi Info 2000 data is stored in Microsoft Access files for maximum compatibility with other systems, many other file types can be analyzed, imported, or exported.

Epi Info 2000 includes a Geographic Information System (GIS), called *Epi Map 2000*, built around the MapObjects program from Environmental Systems Research, Inc. (ESRI), the producers of ArcView. *Epi Map* is compatible with GIS data from numerous Internet sites in the popular ESRI formats.

Epi Info 2000 retains many features of the familiar Epi Info for DOS, while offering Windows strengths like point-and-click ease of use, graphics, fonts, and painless printing. The programs, documentation, and teaching materials are in the public domain (although "Epi Info" is a CDC trademark), and may be freely copied, distributed, or translated.

Key Features of Epi Info 2000

- Maximum compatibility with industry standards, including
 - ✓ Microsoft ACCESS and other SQL and ODBC databases
 - ✓ Visual Basic, Version 6
 - ✓ World Wide Web browsers and HTML
- Extensibility, so that organizations outside CDC can produce additional modules

- *Epi Map*, an ArcView®-compatible GIS
- Logistic regression and Kaplan-Meier survival analysis
- Teaching exercises
- Entirely new, not just a "port" of Epi Info for DOS
- Microsoft Windows 95, 98, NT, and 2000 compatible
- Allows analysis and importation of other file types

System Requirements

- Windows 95, 98, NT, or 2000, with 32 megabytes of RAM – More for NT
- At least 50 megabytes of free hard disk space
- A 200-megahertz processor is recommended but not required.

Planning an Upgrade to Epi Info 2000 from Epi Info for DOS

- Windows 3.1x users should continue to use Epi Info 6.04. The Version 6.04 “b-to-c update” uses 4-digit years for year-2000 compatibility
- Logistic regression and Kaplan-Meier survival analysis can be used directly with Epi Info 6.xx files
- Conversion programs are provided for moving data files from DOS versions of Epi Info to the Windows version
- For permanent systems, some reprogramming of program (PGM) and Check (CHK) files is necessary.

Technical Support and Further Information

CDC provides funding for the Epi Info Hotline, offering free technical support to Epi Info users during normal East Coast working hours (8 AM – 5 PM, U.S. Eastern Standard or Daylight Savings Time). See the title pages of this manual for phone and FAX numbers and e-mail address.

The Epi Info Worldwide Discussion Group

CDC maintains a LISTSERVer for users of Epi Info. Users who send an e-mail to subscribe to the List will receive an email with instructions in return, and then will automatically receive messages submitted to the List by other users. Typically, a request for advice or opinions on a question related to Epi Info will be answered by several messages from other users. The List is an excellent way to stay in touch with other users. The Epi Info Development Team participates in the List, and announcements of upgrades, bug fixes, or other events are sent via the List. It serves as an important forum for guiding future development of Epi Info.

Downloading Epi Info or Related Programs from the Internet

From the CDC home page at www.cdc.gov, choose “Publications, Products, and Software,” then “Software” or go directly to www.cdc.gov/epiinfo. Click on Download to see further instructions, or browse around the Epi Info pages to learn more about the programs. Epi Info and related materials are available on other Web sites. A search for “Epi Info” with one of the many Web search engines will provide access to these sites.

References

- Burton AH, Dean JA, Dean AG. Software for data management and analysis in epidemiology. *World Health Forum* 1990; 11:75-77.
- Dean AG, Dean JA, Coulombier D, Brendel KA, Smith DC, Burton AH, Dicker RC, Sullivan K, Fagan RF, Arner TB. Epi Info, Version 6.04a, a word processing, database, and statistics program for public health on IBM-compatible microcomputers. Atlanta: Centers for Disease Control and Prevention; July 1996.
- Dean AG, Gerstman BB. Computing and epidemiology. In: Gerstman BB. *Epidemiology kept simple*. New York: John Wiley; 1998. p. 275-288.
- Dean AG, Shah SP, Churchill J. DoEpi: Computer-assisted instruction in epidemiology and computing and a framework for creating new exercises. *Am J Preventive Medicine* 1998; 14(4):367-371.
- Dean AG. A course in microcomputer use for epidemiologists and others who count things, using Epi Info. Atlanta: Centers for Disease Control and Prevention, Epidemiology Program Office; 1994.
- Dean AG. Using a microcomputer for field investigation. In: Gregg, MB. *Field epidemiology*. New York and London: Oxford University Press; 1996. Chapter 12, p. 164-180.
- Dean AG. Epi Info and Epi Map: Current status and plans for Epi Info 2000. *J Pub Health Management and Practice* 1999; 5(4): 54-57
- Dean AG. Microcomputers and the future of epidemiology. *Public Health Reports* 1994;109(3):439-41.
- Dean AG. EPIAID. *Byte Magazine* 1985 October; 10:225.
- Dean AG, Dean JA, Burton AH, Dicker RC. Epi Info: A general purpose microcomputer program for health information systems. *Am J Preventive Medicine* 1991; 7:178-182.
- Dean AG, Fagan RF, Panter-Connah B. Computerizing public health surveillance systems. In: Teutsch SM and Churchill RE. eds. *Principles and Practice of Public Health Surveillance*. New York: Oxford University Press; 1994. Chapter 11, p. 200-217.
- Dean JA, Burton AH, Dean AG, Brendel KA. Epi Map 2.0: A mapping program for IBM-compatible microcomputers. Atlanta: Centers for Disease Control and Prevention; 1996.
- Epidemiology Program Office. *Epi Info users' manual, version 2*. Atlanta: Centers for Disease Control; 1987.

Epidemiology Program Office. Epi Info users' manual, version 3. Atlanta: Centers for Disease Control; 1988.

Epidemiology Program Office. Epi Info users' manual, version 5. Atlanta: Centers for Disease Control; 1990.

Fegan G, Dean AG, Shah SP. Epi Info/Epi Map Internet Site. World Wide Web pages (approximately 60) on the Internet. Epidemiology Program Office, Centers for Disease Control and Prevention, 1996. (<http://www.cdc.gov/epo/epi/epiinfo.htm>)

Harbage B, Dean AG. Distribution of Epi Info software: An evaluation using the Internet. *Am J Preventive Medicine* 1999; 16(4): 314-317.

Stroup DF, Williamson GD, Dean AG, et al. Statistical software for public health surveillance (SSS1) (software manual). Atlanta: Centers for Disease Control and Prevention, Epidemiology Program Office; 1994.

Installation



System Requirements and Installation

System Requirements

The following are needed to install and run Epi Info 2000:

- Microsoft Windows 95, 98, NT, or 2000
- 32 mb of Random Access Memory (RAM)--more is recommended for Windows NT
- 50 mb of free hard disk space
- 200 megahertz processor recommended

Uninstalling Previous Versions of Epi Info 2000

If you have a previous version of Epi Info 2000 installed, you should uninstall it before installing a new version. To uninstall, go to the START Menu on your Windows desktop, click on SETTINGS, CONTROL PANEL, ADD/REMOVE PROGRAMS, and select Epi Info 2000. A dialog box will appear for selecting the uninstall method. Choose AUTOMATIC and click the NEXT button and then the FINISH button to complete the uninstall. You can also uninstall the previous version by running the UNWISE.EXE program found in the EPI2000 directory, or by selecting the ADD/REMOVE PROGRAMS icon from the CONTROLS window under SETTINGS on the Windows START menu.

Desktop Applications

Make certain all desktop applications are closed before installing Epi Info 2000. It is usually not necessary to shut down background programs such as virus checkers.

Installing from a CD-ROM

You may install Epi Info 2000 from a CD-ROM or from files downloaded from the Internet. If you are using the CD-ROM, place the CD-ROM into the appropriate disk drive on your computer. Go to the START menu on your Windows desktop, click on RUN, and enter the location of your SETUP.EXE file (i.e. C:\TEMP\SETUP.EXE). Proceed to INSTALLING EPI INFO 2000.

Downloading Epi Info 2000 From the Internet

Before downloading the Epi Info installation from the Internet, create a temporary directory on your hard disk. In Windows Explorer or My Computer, click on FILE,

NEW, FOLDER, and type a name for the folder. This folder will be the directory where you will download the files. Note: Do not name the folder **Epi2000**; TEMP is a good name.

The Epi Info 2000 installation files may be accessed from the CDC web site by using a web browser such as Netscape Communicator or Microsoft Internet Explorer.

After accessing this website, the names of close to 30 SETUP files should appear in the browser. These files may be downloaded individually by clicking on each file and choosing “Save As” under the FILE menu. Proceed until all setup files are saved into your temporary directory on your hard disk. If you have a fast Internet connection, all the individual setup files may also be obtained by downloading a single self-expanding executable file, ALL_IN_1.EXE. After downloading ALL_IN_1.EXE to your temporary directory, run (execute) it to produce the individual setup files. Installation requires either ALL_IN_1.ZIP or the smaller individual files—not both.

The Epi Info 2000 Manual is included in the Setup files in HTML (browser) formats, but you can also download the Microsoft Word (.DOC) or Adobe Reader (.PDF) versions from the website if desired. They are designed for printing the manual as a paper document.

After saving all the setup files in a temporary directory on your hard disk, go to the START menu in Windows 95, 98, NT, or 2000, click on RUN, enter the location of your SETUP.EXE file (i.e. C:\TEMP\SETUP.EXE), or click the BROWSE button to find it, and then click OK to run the Setup program..

INSTALLING EPI INFO 2000

Whether you are running Setup from a CDROM or from downloaded files on your hard disk, a “Welcome” dialog box will appear. Click NEXT to continue with setup. The “Choose Destination Location” dialog box will appear. Here is your chance to install in a directory other than C:\EPI2000. To do so, click on the BROWSE button and select the drive and directory of your choice. Press the NEXT button; the “Select Components” dialog box will appear. By default, all components are marked for installation. To proceed with a complete installation, press the NEXT button. If you wish to install some but not all components, deselect some by clicking until only those you wish to install are checked. Press the NEXT button to continue to the “Start Installation” dialog box. After the files are installed, the “Add to Desktop” dialog box will ask if you want to install a desktop icon. Select YES to create an Epi Info 2000 desktop icon. Additional information dialogs may appear; click OK when you have read them.

After the first batch of files is copied to your hard disk (by a Microsoft program that installs MDAC—the database components of Visual Basic), you may be asked to reboot the computer. In Windows 95 or 98, there may be a second reboot after installing DCOM. Be sure to allow this to happen. Do not install or run other programs before completing the installation.

Components of Epi Info 2000

Epi Info 2000 has a number of components - *MakeView, Enter, Analysis, Statcalc, Nutstat, Epi Map, VisData*, the Manual, and Tutorials. They can be installed individually or collectively. The full installation includes all components of the software. The VisData utility program is installed automatically with any of the other choices.

Full Installation

The default installation will install the all components of the software. When the “Select Components” dialog box appears during the installation process, make certain all components are selected in order to perform a complete installation.

Custom Installation (Selecting components to install)

If you would like to customize the installation by selecting partial components of the software, you may do so when the “Select Components” dialog box appears. Select the options you would like to have installed by using your mouse to select or deselect the desired programs. The dialog box will specify the disk space required and disk space remaining on your computer. The disk space information will reflect the choices you have made.

Drive and Path

Epi Info 2000 will be installed on your C drive unless you choose otherwise. A directory (C:\EPI2000) will be created when the software is installed. If you wish to select another drive or path, you may do so when the “Destination Folder” dialog box appears during the installation process. Use the BROWSE button to choose another location.

Creating a shortcut to Epi Info 2000

When the “Add to Desktop” dialog box appears, select YES to create a shortcut to Epi Info 2000 on the Windows desktop. The icon makes it easy to run Epi Info.

Running Epi Info 2000

After installation, you may open Epi Info 2000 using the desktop icon, or from the START menu on the Windows desktop under PROGRAMS. Epi Info 2000 can also be opened by running the EPI2000.EXE file in the EPI2000 directory.

Getting Acquainted with Epi Info 2000

To become acquainted with the Epi Info 2000 programs, we recommend using the GUIDED TOUR and other parts of the on-line MANUAL. To access the GUIDED TOUR, go to the Epi Info 2000 Menu, and click on the GUIDED TOUR button. The GUIDED TOUR provides specific instructions for using Epi Info 2000 programs. If you are not reading from a paper copy of the manual, we recommend that you print the GUIDED TOUR document and use it as you proceed through the programs. If you have a printed copy of the MANUAL, the GUIDED TOUR is one of the chapters.

The entire MANUAL is available from the main menu of Epi Info 2000. It provides an overview of the features and programs available in Epi Info 2000. The on-line MANUAL is a valuable resource for obtaining information about particular commands or programs in Epi Info 2000.

Installing Translations

After installation, a program called TSETUP is available from the Epi Info 2000 menu under LANGUAGE | INSTALL TRANSLATION. Running this program allows you to install files necessary for running Epi Info 2000 in a language other than English. An example of the Spanish translation is provided.

Precautions

Database (.MDB) files that came with the system, such as SAMPLE.MDB and REFUGEE.MDB, are removed or replaced when uninstalling or reinstalling. If you have inserted any of your own data in these databases (not a good idea), be sure to rename or make backup copies of these MDBs before running SETUP. Databases with other names will not be removed by SETUP.

Problems with Installation

If one or more of the programs is not operating, we suggest uninstalling the software using the UNWISE.EXE file in your EPI2000 directory and then reinstalling after closing all desktop applications. To remove additional uncertainties, it may be advisable to turn off virus checking and other background processes before reinstalling. Reinstall Epi Info 2000, being careful to permit rebooting when it is suggested by the setup program.

If additional problem solving is necessary, the file called INSTALL.LOG, which is written in the installed directory, contains a list of the modules installed and may provide clues to diagnose a problem. You can examine INSTALL.LOG yourself in the word processor or send a copy to a technical support person. Do not edit or erase INSTALL.LOG after installing Epi Info, however, as it is used by the uninstall program to remove files from the computer.

What's New?



Epi Info 2000 Compared with Epi Info for DOS

Features

Windows[®] 95, 98, NT[®], and 2000

The Windows operating systems provide the graphical interface, point-and-click operation, and other features long familiar to Macintosh users. A variety of fonts, graphics programs, and word processors encourage publishing output in many forms to focus on results rather than on programming or processing. When things go well, Windows programs are generally easier to learn and use than are DOS programs. When things are not going well, the additional complexity of Windows programs can make it hard to correct problems, and some users with well-developed typing skills may still prefer DOS.

Windows offers a number of mechanisms for interaction, such as left mouse clicks to activate a control and right mouse clicks to bring up a menu of choices. The familiar FILE, EDIT, and VIEW menus are found in most Windows programs, and serve to make a Windows user feel at home in a new program almost at once. Since users of Epi Info for DOS have already acquired reflexes that do not always match the Windows way of doing business, we have tried to compromise with Windows standards in some cases, to make Epi Info 2000 as familiar as possible to previous Epi Info users.

In *Analysis*, for example, commands all have dialog boxes designed to produce action by making choices and clicking on an OK button. At the same time, however, program statements in plain text are produced and placed in a program editor so that they can be viewed, edited, or run as PGM files. A user can also type the commands directly in the program editor, although the syntax is somewhat more complicated than in Epi Info for DOS because Windows offers more choices.

Visual Basic

Epi Info 2000 programs were developed in Microsoft Visual Basic[®], Version 6. Visual Basic is a computer language used by more than three million programmers worldwide. Since it uses the Microsoft C++ compiler behind the scenes, the resulting programs are as fast and responsive as those produced in any language. Epi Info users do not need to know or use Visual Basic, but advanced users can produce add-on modules for Epi Info using Visual Basic or other popular programming languages. Because Visual Basic is

frequently upgraded by Microsoft Corporation to keep up with changes in Windows systems, Epi Info 2000 will be able to take advantage of new developments in computer technology as they emerge.

Microsoft Access File Format

Databases in Epi Info 2000 are Microsoft Access (MDB) files, and therefore compatible with the most popular Windows database format. Epi 2000 uses the Microsoft Jet Engine (Version 3.51) to manage databases and to read and write a variety of other file formats such as dBASE, Paradox, ODBC, text files, and even HTML tables. Files from Epi Info 6.xx can be imported or exported, but are not read or written directly by the Jet Engine.

Epi Info 2000 databases consist of View tables in a format unique to Epi Info 2000, and data tables in generic Access file format. Because Epi Info and Access use different methods for data entry, it is not recommended that both Microsoft Access and Epi Info be used to enter data in the same database. *MakeView* contains functions for producing Epi Info Views from Access data tables.

Questionnaire Features

Questionnaires in Epi Info 2000 are called “Views.” As in previous versions, a database is constructed by developing screen representations of forms or questionnaires from which Epi Info constructs the database. A program called *MakeView* is provided for developing Views/Questionnaires. Development of a question or data entry item starts with a right mouse click on the screen. This brings up a dialog in which the user can enter the prompt or question and the type of field desired, and choose from a number of properties that, in previous versions, required writing Check code. Legal values, automatic coding, and comment legal values can all be set up from this field dialog.

New features in Epi Info 2000 include multiline fields capable of receiving very large amounts of text, option buttons, check boxes, image fields, and the ability to embed a grid or table in a View. The grid offers an automatic method of creating a related file for which Epi Info 2000 creates and maintains appropriate keys automatically. Grids are particularly useful for repeating values in a questionnaire—age, sex, and immunization status of a number of children on a household form, for example.

Views can have many pages. Each page can have a background image and/or color, and can be assigned an optional name. Moving from page to page is done automatically, by clicking a page name or number in a visible list, or with the PgUp and PgDn keys.

Related views are represented as buttons on their parent views. Clicking on the button brings up the View for the related table. Conditions can be specified so that the button for a related view is only available when appropriate. A hepatitis form might only be accessible if the Disease field has the value “Hepatitis,” for example.

Check code can be inserted in any field to implement IF statements, ASSIGN values to variables, GOTO another field, enforce particular rules, or perform calculations during entry or exit from the field.

Data Entry Features

During data entry, field properties such as REPEAT, READ ONLY, and REQUIRED entry are enforced. Legal and coded fields have dropdown lists of values. Multiline fields scroll automatically when filled with text, and plain text fields scroll to hold up to 128 characters.

Data values entered on a page are saved automatically when the page is completed. First, last, previous, or next records are accessed by clicking buttons on the lower left corner of the screen. For compatibility with Epi Info for DOS, the F7 key moves to the previous record and F8 to the next.

The values for Yes/No fields can be set to any desired screen representation to allow for use in non-English languages, although the underlying values are stored as 0 for “No” and 1 for “Yes” so that data files can be exchanged internationally. The Options entry on the menu can be used to set the values of Yes and No.

Internet Features

The Internet offers a standard page description language (HTML) and two major systems for displaying and accessing documents—the Netscape and Microsoft browsers. To provide for future advancements and prepare the way for even more Internet compatibility, Epi Info 2000 produces output in HTML format. Lists, frequencies, tables, and other results from *Analysis* are in HTML that is compatible with either major browser. Because there are passionate adherents of one browser or the other, or of no browser at all, Epi Info 2000 provides a freeware, off-line browser called Andante4 to display results. Andante4 is small and uncomplicated, and does not access the Internet, but will run an Internet browser automatically when needed if one is installed in the computer. It does not, however, violate any institutional rules against having an Internet browser installed.

The Epi Info 2000 menu can be programmed with the EXECUTE command to access Internet sites automatically, if an Internet browser is present in the computer. EXECUTE `www.cdc.gov\epo\epi\epiinfo.htm`, for example, will bring up the Epi Info home page on the CDC web site from a user-configured menu.

Logistic Regression and Kaplan-Meier Survival Analysis

Two Windows programs written in C++ are provided to do logistic regression and Kaplan-Meier survival analysis. “MVAWin” and “KMWin” are patterned after MULTLR and KMSURV, DOS programs from Dr. Eduardo Franco and Nelson Campos-Filho. The Windows versions read Epi Info 6 files. When used with Epi Info 2000, the necessary files are produced automatically by *Analysis*.

Both conditional and unconditional logistic regression are offered, and KMWin provides several graphs of survival curves. Settings can be altered within the programs and saved for future use.

Geographic Information System (GIS)

An entirely new version of *Epi Map* is included with Epi Info 2000. The core of the program is the same mapping engine that is used in ArcView, a popular Geographic Information System from Environmental Systems Research Institute, Inc. (ESRI). Although *Epi Map* does not have all the features of the high-end commercial programs, ArcView and ARC/INFO, it is capable of reading many of the same file formats, allowing users to tap the enormous mapping resources that are offered on the Internet as SHAPE (.SHP) files. A catalog of resources is included with Epi Info 2000 to illustrate how these can be found.

Epi Map 2000 can display Shapefiles in multiple layers. Each layer can be linked to an Epi Info 2000 data table containing geographic names or codes for the entities of the map. Data for each entity (a count or rate, for example) can be displayed in color/pattern (choropleth) maps as in *Epi Map* for DOS, or as dots randomly distributed within the map's polygons.

New features in *Epi Map 2000* allow displaying streets and placing symbols by coordinates. Thus, one can produce a map similar to John Snow's original map of cholera cases and water-pump locations in London.

Flexible Graphing

The GRAPH command in *Analysis* offers many types of charts for displaying the values of one or more fields in a data table. A toolbar within the graphing module can be activated to allow customization of the resulting graphs. Settings can be saved as templates and used again from *Analysis*.

A Standard Interface for Statistical Modules

Statistical modules as well as the mapping and graphing programs communicate with other Epi Info programs via a standard interface called the "Broad Street Library of Statistics" or IEPI interface. Methods are specified for sending data to the statistical programs and for receiving the results of calculations in the form of HTML pages or as arrays giving the names of parameters and their values. It should be possible for anyone familiar with Visual Basic to write a new statistical procedure to be used with *Analysis*.

Interactive Teaching Exercises

Teaching exercises are provided in HTML format and are available from the main Epi Info 2000 menu under Tutorials. The Oswego foodborne exercise has been expanded to include both data entry and analytic components. The Rhodococcal Infection exercise provides roughly the same components in a hospital outbreak setting. A surveillance exercise provides a basis for planning computerization of a surveillance system. It can be used for classroom discussion and does not have hands-on computer components.

Translation Features

Translation of Epi Info programs, help files, and exercises into non-English languages is made by placing all English phrases used in the programs in a database called

LANGUAGE.MDB, located in the ENGLISH subdirectory under the main directory of Epi Info 2000 (usually \EPI2000). Help files and exercises are also contained in the ENGLISH directory or its subdirectories. Translated phrases, help files, and exercises are placed in similar directories named after the relevant language, such as SPANISH. Hence, translation can be done without changing the names of files, and individual translations can be installed or uninstalled without affecting the main programs. Switching from one language to another can be done from the main menu, and is mediated through setting the LANGUAGE variable in the file EPIINFO.INI.

Converting Epi Info 6.xx Systems to Epi Info 2000

When is it Wise to Convert?

New investigations, surveys, and studies can be done in Epi Info 2000 immediately after its final release. As with any new program, it is a good idea to question any unusual findings and to check them against those from other programs. Any difficulties should be reported to the Epi Info Hotline (see title page of this manual for phone number and e-mail address).

Approaches to Conversion

For systems already developed in Epi Info for DOS and working satisfactorily, it may be practical to retain the previous programs and start by converting the menus in the program to Epi 2000 format. Because the Epi Info 2000 *Menu* will read MNU files from Epi Info 6 with few or no changes, this is a quick way to produce better looking programs with very little effort.

Data (.REC) Files

Converting Epi Info 6 data files and screen formats to Epi Info 2000 is done with the IMPORT entry on the FILE menu of the *MakeView* program.

Check (.CHK) Files

Epi Info 6 check(.CHK) files can be imported in *MakeView* but are commented out prior to manual review and testing, as not all the Epi Info 6 check code is the same or even necessary in Epi Info 2000. Many commands in Epi Info 6 Check code (e.g., RANGE, Read-only status) have become properties that can be selected while creating a field. Making a field READ ONLY, or REPEAT, for example, is done by checking a box in the dialog for that field rather than by writing a command. The DIALOG command allows construction of dialogs for user interaction that should remove the need for some complicated methods that clever users employed to work around this problem in Epi Info 6.

Although the syntax of Epi Info 2000 Check code is somewhat different from that of Epi Info 6, most of the commands are quite similar. Conversion should start by importing the Check code using the IMPORT EPI 6 CHK FILE entry in the *MakeView* FILE menu. The asterisks at the beginning each line can be removed in the Epi Info 2000 program

editor while the code is being converted and tested to run in Epi Info 2000. Errors that occur on running the code can then be corrected by changing the syntax as specified under individual commands in the COMMANDS section of the manual.

Analysis (.PGM) Programs

Analysis programs are generally similar in the DOS and Windows versions of Epi Info, but there are important differences that require changes. One difference is that Epi Info 2000 databases are no longer simple files, but consist of tables within MDB files, and changes in syntax, for example, allow specification of both the MDB and the table name in the READ command. Since each command has a dialog box that generates the command after information is supplied, it should be relatively easy to get the syntax right.

PGM “files” are normally stored in a table inside the MDB where the data reside. This makes a neat package of data and programs that can be copied to another system or sent by e-mail for use elsewhere. Programs can also be saved to or loaded from text files with the extension .PGM, as in Epi Info for DOS. PGM files from Epi Info 6 can be opened in the Epi Info program editor and then edited to conform to the syntax of Epi Info 2000 commands in cases where the syntax differs.

Systems with Related Files

The management of related files differs between Epi Info 6 and Epi Info 2000. Epi Info 2000 creates unique keys automatically for related records and does not require that these keys be defined by the user. This means that importing Epi Info 6 related files into Epi Info 2000 requires converting keys to UniqueKey and Fkey (for Foreign Key) fields. Details are given in the chapter on Data Management.

Helping Epi 6 and Epi 2000 Systems to Coexist

Epi Info 6 files can be imported using *MakeView* and exported from *Analysis* using the WRITE command, but it is not possible to enter data directly into Epi Info 6 files from Epi Info 2000. Hence, a one-time conversion from one system to the other is easiest to maintain. If a single file is needed for analysis by both systems, dBASE files can be used, since they can be read directly by *Analysis* in both Epi Info 6 and Epi Info 2000. *Analysis* reads Epi Info REC files by creating a temporary Microsoft Access file. Permanent conversion can be done by using IMPORT on the FILE menu of *MakeView*.

Gradual, piecemeal conversion of previous systems can be done, linking both Epi Info 6 and Epi Info 2000 programs to the same menu, as long as the data entry program for the correct program is used for entry into the corresponding DOS or Windows files.

The Epi Info 2000 Programs



An Overview of Program Functions

Using Epi Info 2000

Epi Info 2000 is a database and statistics program for public health professionals. Although it can be programmed to produce systems for repeated or permanent use, it can also be used interactively for rapid questionnaire design, data entry, and analysis during an investigation.

Setting up a new database in Epi Info 2000 is done in *MakeView*, the View or questionnaire designer. *MakeView* allows any number of fields or questions to be placed on successive pages of a questionnaire or form. Related Views can also be constructed using buttons for point-and-click access. Legal values, automatic code fields, and more complex data-entry operations can be included in the View at the time it is designed or later.

Once a View is designed, it is saved automatically. A menu choice called ENTER DATA or simply running the *Enter* program with the new View will automatically construct another table in the same database (MDB) for storing the data, and present the form ready for data entry.

During data entry, the special constraints selected during design become active, and the user can move from record to record in several ways, or search for records having particular values. Records are saved automatically whenever a new page or record is requested.

Access to data in Epi Info Views or data tables, or in a variety of other file formats, including HTML and those with ODBC drivers, requires only the READ command in *Analysis*. Output in HTML format is produced by LIST, FREQUENCY, TABLES, and several other commands. Data cleaning can be performed with the aid of RECODE, IF, and SELECT statements. The MAP and GRAPH commands allow data to be linked to industry standard Shapefiles for mapping and to be represented in a variety of graph types.

A special purpose program called *NutStat* compares data on height, weight, age, sex, and arm circumference with international reference standards for assessment of nutritional status. *NutStat* can be linked to an Epi Info 2000 View for data entry or used alone.

The *Epi Map* program allows a variety of maps to be constructed and linked to data in Epi Info or Access data tables. *Epi Map* offers a high degree of compatibility with the popular ArcView program of the Environmental Systems Research Institute, Inc. (ESRI).

Epi Info 2000 can be downloaded from the Internet or obtained on CD-ROM from a variety of sources. Using the programs requires that they be installed on a computer or network under Windows 95, 98, NT, or 2000 using the *Setup* program provided. See the Guided Tour chapter for specific installation instructions.

This chapter provides a description of the capabilities of each of the Epi Info programs. The Guided Tour chapter gives examples of each of the major functions. It is recommended that you follow the instructions on the Guided Tour to become familiar with Epi Info 2000 whether you are a new user of Epi Info or have experience with the DOS versions of the programs, since many of the features of Epi Info 2000 are different from those of Epi Info, Version 6, for DOS.

Setup: The Installation Program

Windows programs require a special installation program that analyzes the type of Windows software already on a particular computer, copies files onto the hard disk, and then registers modules that require registration. According to the usual practice, this program is called SETUP.EXE in Epi Info 2000. It offers choices of various modules to install and locations for installation. After installation, a second program offers to install available translations so that the programs can operate in languages other than English.

During installation, a list of steps called INSTALL.LOG is created so that the UNINSTAL (actually UNWISE.EXE) program can be used to remove Epi Info 2000 from the computer, if desired. Since uninstalling removes programs that may have been placed in COMMON or SYSTEM directories and cleans up the system registry, it should always be used for uninstalling rather than merely deleting files from the hard disk. UNINSTALL will not remove files that are created after installation, such as database files, but they should be created with names other than the SAMPLE.MDB and NUTRI.MDB that are supplied with Epi Info, as these files will be removed by uninstalling or copied over during installation.

After installation, a program called TSETUP is available from the Epi Info 2000 menu under LANGUAGE | INSTALL TRANSLATION. Running this program allows you to install files necessary for running Epi Info 2000 in a language other than English. An example of the Spanish translation is provided. Those wishing to become translators should consult the chapter on Translations for further instructions.

See the Installation chapter for detailed instructions on using *Setup*.

The Epi Info 2000 Menu: A General Purpose, Configurable Menu

The menu provides access to the other Epi Info programs, but is also a general purpose, configurable menu that can be used in Windows 95, 98, NT, or 2000 to offer menus and command buttons that lead to other programs, functions, images, and text to create a theme. By choosing menu items or buttons on a menu, the user can run other programs

or any function for which a Windows or DOS program is available. A command called EXECUTE can be used to allow Windows to determine from the file extension (.HTM or .TXT, for example) which program should be used to display the file. Because the menu can run DOS batch file commands, a series of operations, such as copying files for backup or transmission to another destination, can be specified for a menu choice or button.

The menu items and commands to be run are determined by a text file with the extension .MNU that can be created or edited in the WordPad program that comes with Windows. Menu files can also be created in Microsoft Word or Corel WordPerfect, but be sure to save the file as a DOS text file and confirm that it has only the .MNU extension.

More than one menu can be created to form a series of connected menus. Each menu can have a different image, set of menu items and corresponding functions, and a language translation database for easy implementation in non-English languages. The global variable LANGUAGE can be set from a menu choice to indicate which translation tables should be used by both the menu and the other Epi Info programs. The SETTINGS menu item can be used to specify a Working Directory; the computer is logged into the specified directory before a program is executed from the menu.

A guided tour of the *Menu* program is provided.

MakeView: The Questionnaire and Form Designer

Questionnaires and forms in Epi Info 2000 are called Views. The *MakeView* program is used to place prompts and data entry fields on one or many pages of a View. Since this process also defines the database that will be created, *MakeView* can be regarded as both the form designer and the database design environment. Right-clicking the mouse establishes the location of a field. The prompt or question, type of field, and other details are then entered in a dialog box. Special features such as legal values, codes, ranges, repeat fields, and read-only status are specified in this field dialog. The field can be moved by left-clicking and dragging the prompt, and its size can be changed by clicking on the field to bring up sizing bars.

Data or field types are available for text, numbers, dates, yes/no, and other types of data. Special field types include multiline text fields that can hold large amounts of text; grids or tables that automatically result in related files for repeating groups of values, such as data on children in a household questionnaire; and images.

Questionnaires can have as many pages as desired; additional data tables are created and linked automatically if the number of fields exceeds the limit of 255 imposed by the Microsoft Access file format. Images can be placed in the background of a View page, and a background color can be specified. The font and size for each field can be set separately.

Check code instructions can be inserted for any field to do calculations, perform quality control, or provide advice to the user. Temporary variables can be defined and assigned values from fields on the screen or other defined variables. IF statements allow

conditional operations, such as skipping part of the questionnaire, under specified conditions. Check code can be executed either as the cursor enters a field or as it leaves the field after data entry.

Related Views can be constructed. These are linked to a parent view automatically by unique keys generated by the system. A related view is made accessible through an Internet-style button; the button can be designed so that it is available only under specified conditions—when additional information is needed about a particular disease condition, for example.

Functions are provided for importing files from Epi Info for DOS, for aligning fields, and for placing a layout grid on the screen. Logical groups of fields can be chosen for display on panels so that, for example, SYMPTOMS are displayed as a group and can be referred to by the group name in *Analysis*. In *Analysis*, the command `FREQ SYMPTOMS` would then produce frequencies of all the fields displayed in the Symptoms group.

A guided tour of *MakeView* is provided.

Enter: The Data Entry Program for Windows

The *Enter* program displays the View constructed in *MakeView*, constructs a data table if necessary, and controls the data entry process, using the settings and Check code specified in *MakeView*. The cursor moves from field to field and from page to page automatically, saving data as necessary. Buttons provide access to new, previous, next, first, and last records, and to related tables.

A menu item allows the user to set the values for display of Yes and No. The values actually stored in data tables are 0 for No, 1 for Yes, and 2 for Unknown, but *Enter* and *Analysis* can display any equivalents, such as “True” and “False” or “Si” and “No” that the user specifies.

When related files are accessed with their corresponding buttons in *Enter*, the main form can always be brought up with the HOME button and the parent of the current form can be reached with the BACK button. Navigation thus resembles that for web pages on the Internet as much as possible.

A search function is provided so that records can be located that match values specified for any combination of variables.

A guided tour of *Enter* is provided.

Analysis: Cleaning, Transforming, and Analyzing Data to Produce Tables, Maps, and Graphs

The *Analysis* program provides access to existing data either directly or through Views. It is able to read data from files and tables created in Epi Info, Microsoft Access, FoxPro, dBASE, Paradox, and any other format for which an “ODBC” driver is available. Since SAS, Oracle, Informix, and many other databases have ODBC drivers, it is possible to read a wide variety of data sources. Most other programs can produce one of the supported formats, or at least a fixed-field or delimited text file that will be read by

Analysis. *Analysis* can even read HTML tables, allowing data to be extracted directly from Internet Web pages.

The Microsoft Access 97/Epi Info 2000 format is standard for *Analysis*. Other types of files are accessed through “Links” created in an Access database. If you choose to read a file format other than ACCESS/EPI2000, you will be asked to specify a new or existing Access database file (.MDB). When you choose the data file, a link is placed in the specified MDB file that can be accessed thereafter as an Access data table. Although it actually contains only information on the type and location of the non-Access file, it serves as a proxy for that file even if it is on a network or in another computer. When an Epi Info 6 file is read in *Analysis*, a temporary Microsoft Access file is created automatically and then later removed.

Analysis has commands like READ, LIST, FREQ, TABLES, GRAPH, and MAP. The commands are displayed at the left edge of the screen. Clicking on a command brings up a dialog so that you can generate the command by making choices from the dialog. You can also type the command directly or save it and run the resulting program later (as in Epi Info for DOS). The dialog boxes make it easy to get the right syntax for the commands.

After having read a file or data table, you will want to see its contents using the LIST command, and count the instances of each value of a variable using the FREQUENCY command. If you READ the view called viewOswego in SAMPLE.MDB, for example, a frequency of the ILL variable will reveal 46 records with ILL equal to “Yes” and a total of 75 records. A cross tabulation of ILL by another variable, using the TABLES command, will produce a 2x2 table with the count for each combination of the two variable values (if the variable is also a yes/no variable). Frequency tables are accompanied by confidence limits on the proportions, and 2x2 tables by odds ratios, risk ratios, and several types of confidence limits on these ratios, as well as chi square and Fisher exact tests. Stratified analyses result in Mantel-Haenszel summary odds ratios and confidence limits. For continuous variables like blood pressure or height, the MEANS command provides one-way ANOVA and Student’s t-tests, plus the Kruskal-Wallis non-parametric test in case the former are inappropriate.

Output from statistical analysis consists of Internet-compatible files in Hypertext Markup Language (HTML) that can be displayed in any browser. An index called the RESULTS LIBRARY is automatically generated when a new output file is created. A small offline freeware browser is supplied with Epi Info, and no browser or Internet connection is required to view the results.

Linear regression and logistic regression can be performed. Logistic regression is done in a separate program that also analyzes files in the Epi Info for DOS format. It can be run from *Analysis*, however, after choosing the variables to be included in the mode. A companion program for Kaplan-Meier survival analysis is provided for analysis of longitudinal studies, such as treatment trials with varying degrees of follow-up.

Analysis offers GRAPH and MAP commands with commonly used options. Both maps and graphs can be customized and saved in .MAP or Chart (.CHT) files that can be recalled later to reproduce the custom graph or map with a different data set.

Epidemiologic data in the field is rarely “clean,” as it may be in a textbook or teaching laboratory. The cleaning process in *Analysis* is facilitated by commands to DEFINE new variables and ASSIGN values using a variety of mathematical and logical functions. IF statements provide conditional operations and RECODE allows grouping of data for age and other variables, or transformation of coding from one system to another.

Analysis commands can be saved in program (PGM) tables or in text (.PGM) files that can be run repeatedly or at a later time. Programs can be linked to an Epi Info menu to provide processing options for complete systems, such as public health surveillance systems.

A guided tour of *Analysis* is provided.

The Windows Word Processor

The Epi Info 2000 menu provides access to the Wordpad word processors supplied with all Windows operating systems. By editing the Epi2000.MNU file, you can change the word processor associated with the menu choice or button by that name. To do this, first make a backup copy of your Epi2000.MNU file under another name. Then, edit the block of commands in the MNU file called WordProcess. Between the words BEGIN and END, you will find an EXECUTE statement. Edit the EXECUTE statement to contain the correct path and name of your word processor. The statement should be the same as one that will run the word processor from the RUN window of the Windows START menu. For example, "WinWord.exe" may work for Microsoft Word. Save the Epi2000.MNU file as a text file and test the menu to see that it runs your word processor.

Troubleshooting Hint: A common cause of problems in editing an MNU file is that some word processors add an extra extension when saving to a text file, resulting in Epi2000.MNU.TXT. The latter extension must be removed by renaming the file in Windows Explorer or My Computer before the Epi2000 menu will work with the MNU file.

NutStat: The Anthropometric Calculator

NutStat compares a child’s age, sex, height, weight, and arm and head circumference with reference growth curves from either of two sources, and calculates percentiles, z-scores (number of standard deviations), or percent of median, as well as body mass index. Graphs of the growth standards with or without values from a particular child or group of children can be displayed or printed.

For height, weight, and height for weight, *NutStat* offers a choice of reference growth curves from the 1977/1985 CDC/WHO International standards. To use *NutStat*, choose it from the menu, use the FILE menu to OPEN a project database and a suitable table, and enter the age, sex, height, weight, and other optional characteristics of a child. Extensive configuration options for including or excluding items in both the English and Metric systems are available from the CUSTOMIZE option on the FILE menu.

NutStat will import data already available in Microsoft Access files, and perform calculations during the importation, storing the results in the *NutStat* file format. *NutStat* will also perform calculations on an Access file and add the results to fields specified by the user within the original file.

In an Epi Info View, the RELATE button in a *MakeView* field dialog can be used to create a button and a relationship with *NutStat*, so that the nutritional items can become part of a larger questionnaire--a clinical history form, for example.

A guided tour of *NutStat* is provided.

Epi Map: A Mapping Program Compatible with Popular GIS Programs

Epi Map, the mapping component of Epi Info, is built around MapObjects software from ESRI, the makers of ArcView and ARC/INFO, popular Geographic Information System (GIS) tools. *Epi Map* displays Shapefiles from these two systems, and thus can use the enormous reservoir of map boundaries and geographic data available on the Internet in ESRI-compatible formats.

Epi Map is designed to show data from Epi Info 2000 files by relating data fields to SHAPE files containing the geographic boundaries. Shapefiles also can contain data on population or other variables, and can therefore provide numeric data that become part of the display either as numerator or denominator. Numeric data can be displayed either as color/pattern (choropleth) maps or as dot density maps with the dots randomly distributed within geographic regions. (To avoid calling geographic regions “polygons,” we will refer to each named polygon as a “region.”)

Point locations can be plotted automatically from data files containing x and y coordinates in various symbols, colors, and sizes. Shapefiles can contain lines or points to represent streets or point locations, and points can be placed on top of the Shapefile layer to represent homes in which cases occurred or other geographic points of interest.

Most of the work in *Epi Map* is done from the Map Manager, which allows Layers to be constructed with Shapefiles and related data variables. Layers can be removed or moved to the front or rear of a series of layers. Many properties affecting colors, shapes, grouping of data, type of display, and other characteristics can be set in the Map Manager. When a map has been completed, it can be saved as a template or .MAP file that can be recalled at a later session or from *Analysis* to reproduce the original map.

A guided tour *Epi Map* is provided.

Visualizing Data: The VisData Program

VisData is a utility program, supplied by Microsoft with Visual Basic, for reading data files and examining and changing their properties. *VisData* can be used to examine the fields and properties in a variety of data file types, to display the data in spreadsheet (grid) format, and to edit or delete data items, rows, or columns. Utility functions are offered to COMPACT or REPAIR data files. COMPACT does not perform compression, but will reorganize files in which a great deal of editing has been done so that they occupy less disk space. REPAIR examines damaged files and repairs the damage if possible.

A guided tour *VisData* is provided

Rapid Tour



For Experts In a Hurry

Epi Info 2000 Menu

Explore the main menu of Epi Info 2000. Using the Word Processor, examine the file EPI2000.MNU, which configures and programs this particular menu. Compare it with the file SURVEIL.MNU, which configures the Surveillance menu that pops up when you choose SURVEILLANCE SYSTEM under EXAMPLES.

MakeView

Run *MakeView*. Choose MAKE NEW VIEW from the FILE menu, and give your first or last name to the database, and “NEW1” as the View name. Then create the first field by clicking the right mouse button where you would like to place a question and responding to the dialog. Create several fields and then use the ADD PAGE button to move to page 2. Be sure to include a grid field and a multiline field. Use the RELATED VIEW button to create a button to access a related View. Move the cursor over the button to see the keystroke combinations to resize or move the button and to go to the related View. Use ctrl-left-click to go to the new View, and then right click to add one or more new fields. A related form is created, and is represented and accessed by a button. When you have completed the view(s), choose ENTER DATA from the File menu to run the *Enter* program and enter data in your new view. If you wish to add Check commands, return to *MakeView* by choosing EDIT VIEW from the File menu and then clicking on the Program button. Choose a field and then add commands with the aid of the dialogs that appear when you choose a command.

Enter

Run *Enter*. From the FILE menu, choose OPEN, the database SAMPLE.MDB, and the View SURVEILLANCE. Enter several records until you are familiar with the program features. Choose Hepatitis B for the DISEASE field in one record and Lyme Disease in another.

Analysis

Run *Analysis*, click on the READ button, and choose the SAMPLE.MDB database and the OSWEGO view. Try LIST, FREQ, TABLES (of VANILLA by ILL) and GRAPH (of AGE). Experiment with the SELECT and WRITE commands to write a new table (or

perhaps a dBASE file) containing only cases (ILL= "+"). READ the new file and verify that you have 46 records only for ILL= "+".

NutStat: The Nutritional Anthropometry Program

Run *NutStat*. Open the NUTRI.MDB database and then the nutChildren database. Choose CUSTOMIZE from the menu and adjust the settings for your locality and measurement preferences. Enter several records, using the NEW buttons at the top to move to a NEW child's ID or a NEW date of measurement for a particular child. Return to Record 101 and click on GRAPH. Set the graph to ZSCORE on the CUSTOMIZE screen and choose GRAPH again.

Epi Map

Run *Epi Map* from the main menu. Choose MAP MANAGER from the FILE menu. Use ADD MAP LAYER to display MXState.SHP, selecting NAME as the Geographic Field. Use ADD DATA to select MXMapOct from the SAMPLE.MDB database, and choose PerAdolBIRTHoct98 as the Data Field. You should see a choropleth (color/pattern) map of "Percentage of Births to Adolescents for October 1998."

To display points from a database, using John Snow's example of the 1854 cholera outbreak in the Soho District of London, choose CLEAR ALL LAYERS and then use ADDMAPLAYER to load the SohoSt.SHP Shapefile. Choose ADD POINTS and then, in Sample.MDB, select the SohoPumps table. Choose X_COORD as the X Field and Y_COORD as the Y Field, click on NAME to produce labels, click on Color to choose a color, and select 15 as the size. You should see the water pumps locations displayed on the map. Now display houses with cases by going through the same process with ADD POINTS and the SohoDead table in Sample.mdb. Size 4 is a good choice for the symbols. You should see the houses with cases and their relationship to the pump locations. Bearing in mind that denominators are an important issue, you can nevertheless appreciate a strong geographic association with the Broad Street pump.

The VisData Utility for Viewing Data

Run *VisData* and choose Open Database and then Microsoft Access. Open the LANGUAGE.MDB database and then double-click on the NUTSTAT table. You should see a grid filled with phrases from the *NutStat* program in six languages. This is a part of the database that allows for easy translation of programs.

Guided Tour



Getting Acquainted With the Programs

Taking the Guided Tour

This chapter provides step-by-step instructions for exercising the main features of each program. If you are reading from the screen, you may find it convenient to print the instructions using the PRINT command on the FILE menu of your browser or word processor. On a large screen, it is also possible to keep the instructions open in one window and work with Epi Info elsewhere on the same screen.

Guided Tour of the Epi Info 2000 Menu

Highlights:

- Configurable features of the main menu
- Example of an entirely different menu constructed by copying, renaming, and editing the Epi2000.MNU file.
- Brief look at an MNU file

The screen image for the main menu of Epi Info 2000 is John Snow's famous map of the location of cholera cases surrounding the Broad Street pump in London in 1846. Artistic license has been taken in representing cases near the pump as vertical solids rather than with Dr. Snow's neat tick marks.

The main programs of Epi Info can be accessed either through the PROGRAMS menu or by clicking on the buttons. The buttons can be turned on or off with the BUTTONS item on the SETTINGS menu. The Guided Tour is part of the Epi Info 2000 manual and help-file system represented on the MANUAL menu. Several exercises for learning epidemiology and computing are found on the TUTORIALS menu.

On the EXAMPLES menu, choose SURVEILLANCE MENU. A second menu appears with different menus, background picture, and screen text. This is the beginning of a Surveillance system to be developed in Epi Info 2000, but most of the entries on this menu are not yet active. The menu is presented to show how easy it is to customize the menu and use it for your own programs. Close the Surveillance menu by choosing EXIT from the ENTER DATA menu.

To gain confidence and to see that your system is functioning well, you might click on each of the buttons on the main menu and briefly examine the program that appears. Exit from each one by clicking an Exit button, the small “x” box in the upper right corner, or

EXIT in the FILE menu. We will visit each program again in more detail, as described below.

If you are interested in the details of the files behind the menu, use the word processor and OPEN the files EPI2000.MNU and SURVEIL.MNU. Note that differences in these text files are responsible for the differences in appearance and function of the two menus.

Guided Tour of the *MakeView* Program

Highlights

- Designing a new form or questionnaire (a View)
- Text and numeric fields
- Specifying a list of Legal values
- Inserting a grid, the automatic way to deal with repeating data within a questionnaire
- Large text (multiline) fields

To run *MakeView*, click on the MakeView button on the main menu screen. You should see a blank page for constructing a “View.” Questionnaires are called Views in Epi Info 2000 because there can be more than one View of a database or data table. A database table with the prefix “view” stores the screen appearance of the questionnaire, the characteristics of the fields, and any Check code that gives special instructions for the data entry process. Data values entered in the *Enter* program are stored in another table, without a special prefix.

To make a view, from the FILE menu choose Make New View. The dialog CREATE OR OPEN PROJECT appears. Enter a name for your project database, such as your name or initials, and click OPEN.

A project or database (.MDB for "Microsoft Database") file can hold as many Views and data tables as you wish (well, up to 1000, anyway). Generally it is best to create a new MDB file for each project you develop, as an MDB containing hundreds of tables will be hard to copy to diskettes. *Analysis* programs for processing the data can be stored in the same MDB with the data, making a convenient project package.

In the "Name the View" dialog, enter MOTHER as the name of the View within the MDB, and then click OK. Place the cursor near the upper left corner of the blank page and click the *right* mouse button. The field dialog box that appears offers options for entering the prompt, the field type and length, and a number of the characteristics that were previously implemented in Check files.

For the first field, enter the prompt “First Name” (without the quotation marks) and press Enter twice. This makes a text field that can hold up to 255 characters.

For the next field, you could move the cursor and right-click with the mouse on a suitable location, but, to see a shortcut method, press Enter instead while the cursor is in the FIRST NAME field. The field dialog pops up again and you are ready to enter “Last Name” as the prompt. After doing so, press Enter twice, and note that the second field is now automatically positioned on the View.

Below First Name, right-click to add another field. Enter the prompt “Today’s Date,” and use the scroll bar to the right of the field types to see the rest of the list of types. Choose the DATE type and the appropriate date format as MM-DD-YYYY or DD-MM-YYYY in the dialog. Click OK. Add another field for “Date of Birth,” using the same field type and pattern. Click OK.

Right-click on the form to make a field for AGE. Type “Age” as the prompt. Choose NUMBER for the TYPE and then choose ### or ## from the PATTERN list. You can also type patterns into the pattern window. Click on OK at the bottom of the dialog.

The next field is “Sex.” We will use it to illustrate how variable names are constructed. Right-click where you would like to place the field. Type “Male, Female, or Unknown Sex” in the prompt window, press Enter, and note what appears in the Field Name window on the right. Now click again in the prompt window, and with the left mouse button held down, select just the word “Sex”. Double-click on the selected word, “Sex,” in the prompt window. Note that the variable name becomes SEX. (In Epi Info 6, we would have enclosed “Sex” in curly brackets.)

Now create legal values for SEX by clicking the LEGAL VALUES button. In the dialog box that appears, choose CREATE NEW, and then enter suitable values (Male, Female, Unknown) in the list that appears, pressing Enter after each to obtain a new blank line. Click OK, and then OK again in the field dialog box. Note the button on the right side of the SEX field. Left-click on the button to show the list of legal values from which to select during data entry.

To move a field on the screen, click on the prompt for the field and drag it to a new location while holding the left mouse button down. Use this method to space the fields on the page. Most types of fields can be resized by clicking in the field and then clicking and dragging the colored “handles” that appear. Text fields are limited to one line, but we will add a multiline field later.

It is time to save the page and add another one. Click on the ADD PAGE button under the page window on the left side of the screen. The first page is saved automatically and a blank page appears.

We are going to insert a Grid (table of columns and rows) on Page 2 to record the names, ages, and immunization status of the children in the household. Right-click in the upper left corner of the form and enter “Children in the Household” as the prompt. Click on the INSERT GRID button in the dialog. Enter the name of the first grid column, “Name,” in the prompt box. Click on SAVE COLUMN about midway down the form (DONE at the bottom is now for the entire grid.). Enter the second column as “Age” and make it a NUMBER. Click SAVE COLUMN and enter the last column as a text field called “Immunization.” Now click SAVE COLUMN once more and then DONE at the bottom of the dialog. The grid will appear on the View.

Click on the grid so that handles appear around it. Click and drag the lower right handle or others to adjust the size of the space for the grid. Click outside the grid to remove the handles. To adjust the size of the columns, hover the mouse over the line between two

column headings until a right/left arrow appears. Then hold the left mouse button down and drag the line to the right or left to adjust the column width.

The grid will create a related file that will allow the user to enter as many children as needed for each household. The program automatically maintains an appropriate key for linking the related file.

Click ADD PAGE again and you are ready to make page 3 of the View. Now make a field of the MULTILINE type having the prompt, "Comments of Interviewer." Click OK and then click on the field, adjusting its size as you did with the grid to make it big enough to enter a number of comments. There is no practical limit on the amount of text that can be entered in MULTILINE fields. Add a text field for "Interviewer's Initials," and save this page with the SAVE command on the FILE menu.

Checking and Controlling Data Entry

Highlights

- Inserting commands to customize the data entry process
- Calculating an age from two dates

We would like AGE to be calculated automatically after entering Today's Date and a Date of Birth. If both of these dates are given then the cursor should skip over AGE after the calculation is performed. These functions can be programmed in the Check code environment.

Return to page 1 by clicking on its entry in the page list on the upper left. Bring up the Check environment by clicking on the PROGRAM button to the left of the view. A list of fields and available Check commands appear at the top of the screen in a tabbed dialog, and the program editor is displayed below.

Click on the arrow under "Choose Field Where Action Will Occur" to pull down the choices. Since we want Age to be calculated after DateOfBirth is entered, choose the DateOfBirth field. Click on the VARIABLES tab and then the ASSIGN command. Display the list of available variables for the ASSIGN VARIABLE by clicking on the arrow. Choose AGE, and then type *Years(DateOfBirth, TodaysDate)* in the next space. Choose the field names from the pull-down list if you do not know them in advance. Click on the SAVE button and note that the command,

ASSIGN Age=Years(DateOfBirth, TodaysDate)

appears in the program editor. YEARS is a function that calculates the interval between two date fields in years rather than days, weeks, or months. The starting date is listed first and the ending date second in the parentheses, separated by a comma. A list of functions is contained in the Functions and Operators chapter of the manual.

To make the cursor skip over AGE if it has been calculated, choose AGE as the FIELD WHERE ACTION WILL OCCUR. A prompt asks if you wish to save the previous commands. Click the YES button. Now from the RECORDS tab, choose the IF command. In the CONDITION blank, type (or select from the pull-down list and the buttons) the condition, AGE>1. Click on the THEN button. Now from the FIELDS tab,

click GOTO, and within the next dialog, type or select the variable SEX. Click OK, and you should see in the program window, the command,

```
IF AGE >1 THEN
```

```
    Goto Sex
```

```
END
```

You have written your first Epi Info 2000 Check program. Exit from the Check programming facility by clicking on the OK button and answering “Yes” to the prompt about saving Check code. This completes the view. Although you could exit from *MakeView* at this point and run the *Enter* program from the Epi Info 2000 menu, it is more convenient to run *Enter* from within *MakeView*.

Making a Database

Remain in *MakeView*, choose ENTER DATA from the FILE menu and respond “OK” to have the program construct a database from the view. The data table will have the name displayed unless you choose to edit the name. The *Enter* program displays the view for data entry.

Guided Tour of the Enter Program

Highlights

- Entering data and verifying that your age calculation works
- Moving from page to page
- Opening an existing View and database
- Navigating from record to record
- Searching for particular records
- A sample surveillance view with related views available according to disease condition

You should have the MOTHER questionnaire on the screen. If not, go back to the main menu and choose ENTER DATA, OPEN on the File menu, and then the database that you created and the MOTHER view. Enter data in the fields displayed. After you enter Date of Birth, the age should be calculated automatically, and the cursor should jump to the Sex field. At the end of each page, the entries will be saved automatically. On page 2, fill in the grid with reasonable answers. After the first line, a second is created automatically. Enter as many children as desired, moving the cursor with the ENTER key and/or arrow keys. Press the Enter or Esc key to move to the next page, saving page 2 on the way. You can also click on the NEW button to save the current record and move to a blank record.

After entering the data for page 3, press the Enter key. You should now see an empty record 2, ready for entry. Note that the record number appears on the lower left.

Opening Another View

To see another view containing more data, choose OPEN from the FILE menu and then click on CHANGE PROJECT. Choose the database SAMPLE and the View OSWEGO. Note the record number at the lower left is 76, representing the 75 records in the table plus the empty record now on the screen.

Moving From Record to Record

Examine the records in the file by moving from record to record with the arrow buttons on the lower left. The double arrows move to the first and last records; single arrows move one record at a time. To move to a new record, click on the double right arrows twice.

Finding Records

To find records matching specified criteria, click on the FIND button on the left. A dialog box appears. Choose the AGE field and then type “11” (without quotation marks) in the field that appears. Click on the OK button to find all the records in which AGE is 11. To choose one of these records for editing, double-click on the left side of its row until the entire row is highlighted and the selected record appears on the screen. If you prefer not to select one of the records shown, but to continue with the current record, click on the BACK button.

A More Complicated View with Groups of Variables and Related Views

Open the view called SURVEILLANCE in the SAMPLE.MDB database. Note that the variables are arranged on panels. Each panel is a GROUP. In *Analysis*, group names can be referred to as a shortcut to perform operations on all the variables in the group. For example, LIST PersonalInfo would display data from all the variables in the lower group panel.

Related Views are displayed in the box in the lower part of the left panel. Because each of the Views shown is designed for a particular disease, they are inactive and do not respond to mouse clicks when you first open SURVEILLANCE and a new, blank record is displayed.

Use the single and/or double left arrow buttons in the lower left panel to move back to record 1. The double arrow buttons move immediately to the first or last records in the table. The single arrows move one record at a time. The New button moves to the next empty record.

Note that the Disease in record 2 is “Hepatitis,” and that the HEPATITIS DETAILS button is therefore active. Click on this button to see the special form for Hepatitis. Use the Back button to return to the main Surveillance form.

After experimenting with SURVEILLANCE and perhaps entering one or more records, choose EXIT from the FILE menu to return to the main menu.

Creating a Relational Database

Relational databases are created in *MakeView*. To create a new relational database system open *MakeView* and select MAKE NEW VIEW from the FILE option in the pull down Menu. It displays a form called NAME THE VIEW that allows you to enter the name of the new view. Click on CHANGE PROJECT to create a brand new one.

Step 1: Create a New Project

In the FILE NAME box, type a unique name for the project MDB (e.g., your first name) and click on Open. The new project is empty and you will be asked to name the new view to create. Type "House" in the box labeled VIEW NAME and press enter or click OK. There should be an empty form on the screen with the word "House" at the top.

Step 2: Create the main (parent) view

By right clicking on the View and responding to the questions in the field dialog, add the following variables to your database:

Prompt	Variable Type	Pattern
House Id	Numeric	#####
Address	Text	
Address 2	Text	
Fence	Yes/No	
Any Children?	Yes/No	
Comments	Multiline	

Step 3: Create a Relational Button

In the lower part of the page create a new variable. In the prompt field, type "Children" and then click on the button labeled CREATE RELATED VIEW.

Step 4: Set up the relational properties

A form labeled CONDITIONS FOR RELATED FORM TO BE ACTIVE will appear. In the frame labeled FORM SHOULD BE ACCESSIBLE select ONLY WHEN CERTAIN CONDITIONS ARE TRUE (The default value is ANY TIME). There are also two check boxes. They should not be checked. When you select ONLY WHEN CERTAIN CONDITIONS ARE TRUE, a new set of fields is displayed.

Open the box called AVAILABLE VARIABLES and select ANYCHILDREN from the list. Once ANYCHILDREN is selected, it will be displayed on the formula box (labeled FORM CAN BE ACCESSED WHEN...); click on the equal (=) sign and then click on the formula box and add "(+)" without the quotes. The box should read:

ANYCHILDREN = (+)

Click Ok to accept these instructions. The button for the related form will be active only when the value of ANYCHILDREN is "yes."

Step 5: Related view

A new window will allow you to decide if you want to relate your table to a new view or to a view that you previously created. In this case the related view has not been created. Click on OK to accept the default value, CREATE A NEW RELATED VIEW.

Step 6: Move and resize the button

Now there should be a button on the page called CHILDREN, but the button does not work! Click on the button but you will get no response. Place the cursor over the button and wait a minute for the tooltip instruction to appear.

In the *Enter* program, clicking the button will take you directly to the related form, but here in MakeView, things are more complicated. You can move the button by clicking and dragging it to a new location while holding down the *Shift* key. To resize it, click on the button while pressing the *Alt* key. A group of small blue boxes around the button will allow you to resize it. You do not need to keep the Alt key pressed while resizing. Now, to go to the new related View, click on the button while pressing the control (*Ctrl*) key.

Step 7: Create the related view

In the blank page that appears, the title at the top should be CHILDREN instead of HOUSES. In this new view, enter the following variables:

Question or Prompt	Type
Name	Text
Age in Years	##
Sex	Text: Legal (Male, Female)
Sport	Text Legal (Baseball, Football, Soccer, Tennis, Swimming, Other)
Comments	Multiline

Step 8: Save the view

Epi Info 2000 always saves your view automatically, but you can save your work at any time by clicking on SAVE in the FILE option of the pull down menu.

Step 9: Review your work

Two new buttons are displayed on the left side of the form--BACK and HOME. By clicking on BACK you will be taken to the parent table (House). By clicking on the button while pressing the Ctrl key you can come back to the CHILDREN View at any time.

Note: When you click on BACK for the first time, a new window called NEW DATA TABLE will pop up. It will verify that you want not only to create a view but also to create a data table to store information. This form will be displayed only once. Click OK the default values.

Step 10: Testing the new relational database system

From the parent view (HOUSE) click on ENTER DATA located on the File option in the pull down menu. Click OK to create the data table for the main view (HOUSE).

Remember that you will be asked to create the table only once. *MakeView* executes the *Enter* program and you are ready to test your new relational database. Note that at the beginning, the CHILDREN button is not active (grayed out). It will become active only when you enter “Yes” and press enter in the ANYCHILDREN field. Type the data for the first family:

House Id: 0001

Address: 100 any road

Address 2: across the street from the maple tree

Fence: Yes

Children: “Yes”

Comments: “None”

It is possible that *Enter* will not accept “Yes” as possible value for children. In such case, click on options and change the settings to “Yes” and “No” before trying again.

Note that the CHILDREN button becomes active after selecting “Yes” on the ANYCHILDREN field. Now click on CHILDREN button and enter the following data:

Name	Age	Sex	Sport	Comments
John	6	Male	Baseball	
Bob	7	Male	Soccer	
Susana	9	Female	Other	Ballet
Carolyn	8	Female	Other	Gymnastics

Remember to press the “Esc” key to leave the multiline field. When done with the first house, click on “Back” to return to the previous questionnaire. To enter the second family, click on the “New” button.

House Id: 0002

Address: 121 any road

Address 2: in front of the maple tree

Fence: "No"

Any Children: "No"

Comments: "None"

Since AnyChildren = "No" the button is not active so that you cannot enter children for record two. Click on "New" to create a third record.

House Id 0003

Address: 136 any road

Address: 2 next to the school

Fence: "No"

Any Children: "Yes"

Comments: "None"

Name	Age	Sex	Sport	Comments
Mike	11	Male	Tennis	
Amy	12	Female	Baseball	

While entering "Amy" note that the record number is "2" because Amy is the second child for house 0003 while Carolyn was the fourth child for house 1 and there are no children for house 0002.

Guided Tour of the *Analysis* Program

Highlights

- READ a view or a data file or table
- LIST the contents of the database
- Obtain the FREQUENCY of values for a field
- Cross-tabulate with the TABLES command and resulting epidemiologic statistics
- The library of previous output, all in HTML for the Internet
- Choose how "Yes" and "No" are displayed
- Define a new variable and assigning a value
- Use an IF statement to determine and assign case status
- SELECT a subset of records to process
- RECODE values to group the AGE field

- WRITE data to another file or table
- READ a non Access file
- READ related tables in a view and analyze data from more than one table

To run *Analysis*, click on the ANALYZE DATA button on the main menu screen. Note that all commands are shown in the tree view on the left. Clicking on a command will bring up a dialog that places the command in appropriate form in the program editor at the bottom of the screen. Results appear in the third window above the program editor, which is a simplified version of the Microsoft Internet browser.

READING a View in *Analysis*

Click on the READ command. A dialog box appears so that you can choose a database and a view. Choose the project SAMPLE.MDB and the View OSWEGO. Click OK and note that the READ command appears in the program editor in the proper syntax. You are creating a program by responding to questions in the dialog boxes.

Lists

Click on the LIST command. In the dialog that appears, choose one or more variables or click on "ALL" to choose all. Choose GRID as the output format and click OK. The variables are displayed in columns in a scrolling window. Click the button with the "X" in the upper right corner of the Grid to leave the Grid. Try the LIST command again, but choose HTML as the output format. This time the results appear in the form of an Internet web page displayed in the small browser included with Epi Info 2000. This browser displays web pages on the local machine, but is not itself connected with the Internet. If you have another browser on your computer, the Epi Info browser will use it for Internet access if necessary.

Frequencies

Choose the FREQUENCIES command. In the dialog box, use the dropdown menu to select one or more variables, and then click OK. After a short wait, the results should appear in the browser window. Scroll up and down and note that each table is accompanied by yellow bars to the right that indicate the frequencies. Statistics will be displayed below the table if the value of the variable is numeric, as in AGE, but not for Yes/No fields like ILL.

Tables

Click the TABLES command. In the EXPOSURE VARIABLE field, choose VANILLA and for the OUTCOME VARIABLE, choose ILL. This will perform a cross-tabulation of VANILLA by ILL. Note that stratified analyses can be done by inserting the name of the stratifying variable(s). Summary data can be processed by setting WEIGHT equal to the name of a COUNT field. Click OK.

Note that the output in the browser includes a table and a graphic representation of the table values in each cell. Statistics are displayed below the table. If you have a printer

connected, try printing the table by clicking on the PRINT button. (Information on the statistics in Epi Info is available in the Statistics file under HELP on the main menu.)

Viewing Previous Results

Click on CLOSEOUT to close the document you just created and then, click on the hyperlink called RESULTS LIBRARY at the top or bottom of the TABLES output in the browser. An index page appears, showing previous commands that have produced output files. Click on any of the entries to display it. An archiving system is provided so that important results can be selected and saved for future reference. You can learn more about storage of results by choosing the OUTPUT tab and examining the choices under STORING OUTPUT.

Setting the Displayed Values for Yes/No Fields

Click on SET under OPTIONS in the command tree on the left side of the screen. Note the options for customizing output. Change the values to be displayed for Yes/No fields, choosing from those available or typing your own, such as “Si” and “No” in Spanish. Choose LIST again from the STATISTICS tab and verify that the values displayed are those of your choice.

Defining a New Variable

Under Variables, choose the DEFINE command. Type STATUS as the name of a new variable. We want to set this variable to “Case” if the person was ILL and “Control” otherwise. A Standard variable, with the value reset for each record as the program passes through a table, is the best choice for this purpose. Click OK, and the necessary statement will appear in the program window.

An IF Statement

We can use an IF statement to set the value of the new STATUS variable. Choose the SELECT/IF tab and click on IF. The first item in the dialog is the condition under which the following statements should or should not be executed. The final format of the necessary IF statement is:

```
IF ILL = "Yes" THEN
    STATUS = "CASE"
ELSE
    STATUS = "CONTROL"
END
```

You will not have to remember the format, however, since filling in the blanks in the dialog will allow Epi Info 2000 to write the necessary command in the program editor. In the first blank, fill in the Condition as ILL="Yes". To do so, you can choose the variable ILL from the list of variables in the second window and choose the equal sign and condition “Yes” by clicking on buttons with these labels. The “Yes” button will display whatever label is specified for “Yes” in the SET dialog—“(+)”, “Yes”, “Si”, etc.

Regardless of the setting for Yes/No fields, “Yes” is always represented in the database as 1, No as 0, and Unknown as a blank or null value. These values can also be used, as in:

```
IF ILL = 1 THEN      (Note that numbers do not have quotation marks)
```

Etc.

If you are familiar with commands in Epi Info 6, you can type your own command in the box located under the then button. On the other hand, if you are experimenting for the first time with programming languages, you may want to use the aid available for IF statements. In the first case type STATUS="CASE". On the THEN box and STATUS="Control" in the ELSE box.. Click OK to finish the command.

If you are not an expert, click on the THEN button. As you can see the command screen for *Analysis* came in front of the IF form at the top, in the blue bar, you will read “Then Statement” meaning that all commands created will be executed if the condition stated in the IF are true.

Select ASSIGN. A form similar to the regular assign form is displayed, but, in this case, an ADD button replaces the OK button. Click on the arrow labeled “Assign Variable” and select STATUS from the list. On the box labeled “= Expression” type “CASE” (Do not forget to include quotes. Click on ADD to complete the task.

The expression STATUS = “CASE” should appear on the box located under the THEN button. You can edit that statement. Place the cursor between the E and the quote in CASE and add an “S” (Without quotes). The resulting statement should be

```
ASSIGN STATUS = “CASES”
```

Follow the same procedure for the ELSE side of the statement; but in this case, the assign statement would be ASSIGN STATUS = “CONTROLS”

When the IF statement is completed, click on the OK button. The complete IFstatement will appear in the program editor.

Use the LIST command to verify that STATUS is indeed properly set, and that records where ILL = “Yes” have “CASE” as the value of STATUS.

The SELECT Command

The SELECT statement limits subsequent analysis to particular records based on criteria that you specify. To work with cases only, for example, choose the SELECT statement and enter the condition STATUS = "CASE". Use LIST or `FREQ STATUS` to show that cases only are included in the analysis. To return to working with all the records, choose CANCEL SELECT and note that it places the word SELECT, without conditions, in the program.

The RECODE Command

To Group the values of AGE, first DEFINE a new variable called AGEGROUP, using the DEFINE command. Then Choose RECODE and specify that you will recode from AGE to AGEGROUP, using the drop down list. Then enter 0, 18, and “Child” on the first line of the grid and 19, 120, and “Adult” on the second line. Click OK, and then do a LIST to see the results.

Creating a New File with the WRITE Command

At this point, we have made several improvements in the dataset, and might want to create a file containing the new variables. The new file can be either an Epi Info 2000 (Microsoft Access) file or one of many other file types. Choose the WRITE command, and then, from the list of Output Formats, choose “dBASE IV”. Specify “All” variables and “Replace” so that an existing file by the same name will be overwritten. Give your Initials for the file name. Click on OK to write the file.

READING a dBASE File

Now that you have produced a dBASE IV file, it is time to test the flexibility of *Analysis* in READING a variety of file types. Choose READ from the DATA tab, specify dBASE IV format, and read the file you have just produced. Use LIST and/or FREQ to verify that the variables you created and their values are contained in the new file.

READING a View with Related Views

Related views contain different information about a single individual or record and are linked by a common identification variable.. Epi Info 2000 comes with an example of related views called Refugee, taken from a local health department system for following the health status of recently arrived refugees.

Click on READ and then click on the CHANGE PROJECT button. Look for a file called REFUGEE.MDB and double click on it. There are several views available in this project. Click on FAMILY and then OK.

The table called FAMILY contains 539 records. Use LIST (*) to see all fields in the database. ViewFamily relates to ViewPatient by a common field called FAMIDNUM. Click on RELATE. A new window will be displayed In the Views box, highlight the viewPatient table and then click on BUILD KEY. Note that the Build Key button is not active until you select viewPatient from the list.

After clicking on Build Key, a new form appears. It contains two parts “Current Table(s)” and “Related Table.” The current table is active. From the box labeled “Available Variables” select FAMIDNUM and then click on “Related Table”. The variable FAMIDNUM will be shown in the current table(s) box. Now select again from the available variables box FAMIDNUM and click OK. The field FAMIDNUM should be visible in both boxes. Click again on OK to close the form. In the box labeled “Key” there is a new instruction that should say that FAMIDNUM::FAMIDNUM is the relational key. Once you become familiar with these relational keys, you can just type the relation directly into the key box instead of clicking on the Build Key button.

To complete the task, click on OK. As you can see, the header in your output shows now 1772 records and the table is viewFamily join viewPatient.

Create a table using country (located in FAMILY) as the EXPOSURE VARIABLE and sex (located in PATIENT) as the OUTCOME VARIABLE.

Guided Tour of the NutStat Program for Nutritional Anthropometry

Highlights

- Entering data from one child's measurements
- Interpreting results from calculations based on accepted reference standards
- Graphing more than one result to show a child's growth
- Customizing the data entry screen

Run the *NutStat* program by clicking the NUTRITION button on the main menu. The program should obtain the name of the last database accessed from the EPIINFO.INI file and load this database automatically. If you do not see record number 105 and the data table name NUTCHILDREN at the top of the program, use the OPEN command on the FILE menu to open first the NUTRI.MDB database and then the table called NUTCHILDREN.

Click the left arrow next to the IdNumber several times until you see a record belonging to Alouetta Delia. Alouetta has four records in the table. To see the others, click on the left-pointing arrow in the DATE OF MEASUREMENT box and look at each of the earlier records. Click on the GRAPH button. Choose “Z-Scores” as the graph type and click OK to see a graph of her results compared with the International growth reference curves. Age is shown across the bottom of the graph and the left axis shows by how many standard deviations the child's measurement differs from the International average reference standard for that age and sex. The advantage of z-scores (standard deviations) is that both height and weight can be plotted on the same graph. In interpreting the graph, it is important to know that two standard deviations in either direction from 0 is approximately the 95th percentile and three is near the 99th percentile. Hence, in Alouetta's second set of measurements, her height is nearly at the 99th percentile for girls her age, but her weight is slightly below the norm; she must have been tall and slender at age 4 (48 months) when this set of measurements was taken.

To move to another child's record, choose the next IDNumber by clicking on the right arrow in the IDNumber box. By clicking repeatedly, you will discover how many children there are in this database. Click on the second button to the right of the IDNumber to create a new record and IDNumber. Note that clicking the left arrow from the new record will return to the last completed record in the table.

In the new record, enter a name, sex, and birth date. Note that the age is calculated automatically after you press Enter. Now enter values for height and weight (for a 10-year old, 140 centimeters and 40 kilos will do). Note that the statistical calculations appear as soon as you press Enter in the weight field. The body mass index is calculated

automatically (but we are awaiting new standards before calculating the age-specific values for body mass index).

Enter 15 centimeters or 6 inches for arm circumference and note the z-scores. Now click on New in the IDNumber box to save the record and go on to the next.

From the File menu, choose Customize to see the options available. On the Units tab, choose either the English or Metric system of measurement, click on Arm Circumference to deselect it, and then choose OK. Note that the main screen now displays only the measurement units selected and that Arm Circumference no longer appears. Return to Customize and set up the main screen in the configuration you find most useful for your own environment.

NutStat is designed to accept the new NCHS growth reference curves as soon as they become available, but these features have been disabled pending release of the new standards.

Guided Tour of the *Epi Map* Program

Highlights

- Adding a shapefile to create a map
- Adding data to be represented as a color density map
- Creating a map of cholera cases in John Snow's London using X and Y coordinates of case households

First, let's make a map of Mexico representing one numeric result for each State by coloring the polygons that represent the States. Run Epi Map from the main menu. In the FILE menu of Epi Map choose MAP MANAGER, and from the LAYERS card, click on ADD MAP LAYER. You should see a choice of shapefiles (.SHP), containing the boundaries of geographic regions. The SHP extension indicates this is a shapefile, the popular Geographic Information System (GIS) format used by Epi Map 2000 and the commercial program ArcView. Open the file called MxState.shp and choose NAME as the field containing the names of Mexico's states.

A database containing public health data from Mexico is provided. Click on ADD DATA, and then select the SAMPLE.MDB database and choose the table called MexMap95. The appropriate Geographic Field is STATE and the data field is PerTeenBirths95, the percentage of total 1995 births to teen mothers (age <20 years). You should see a map of Mexico with the percentage of all births that occurred to adolescent mothers for each state, represented as a shade of blue. These data were provided by Dr. Fernando Beltrán, and are from the Mexico Ministry of Health, 1995 Vital Statistics Report.

You can experiment with features such as FIND on the EDIT menu (also represented by the binocular button). Try finding "Vera," for example. (This feature is case-sensitive.) The magnifying glasses with + and - signs are for zooming to larger or smaller sizes. Click on the one with the + sign and then click and drag a rectangle over an interesting

part of the map. It will zoom to fill the frame. Try moving the map with the white glove (“panning”). You can restore the size to normal either with the negative magnifying glass or with the world button that represents the global or full extent view. The “i” symbol means “Identify.” Click it and then click on a state to see the data values for that state—both those contained in DBF table of the shapefile and those in the related table of the MDB database.

Under the FILE menu, you can SAVE AS BITMAP FILE, and under the EDIT menu COPY BITMAP TO CLIPBOARD. Once you execute either of these options, you can add or paste the map into documents in other software programs, i.e., reports or graphic presentations. You may want to use the FILE / SAVE AS GIF FILE in preparation for adding the map to a Web page.

Epi Map can represent data as discrete points on the screen, as well as quantitative values within polygons. The following set of instructions will produce a map resembling John Snow’s famous map of cholera cases in the Soho district of London during 1854 in relation to the various public pumps used for drinking water. A popular theory of the time related cases to the presence of an old cemetery where plague victims from a previous century had been buried; hence the cemetery will be included as a separate layer in the map.

First run Epi Map from the main menu. From the FILE menu, choose MAP MANAGER. Choose ADD MAP LAYER, and then open the SohoSt.SHP file. Now you should see a map of the streets of Soho. From the map manager and add the cemetery by using ADD MAP LAYER and choosing SohoBuri.SHP. To add the locations of the water pumps, use ADD POINTS and select SAMPLE.MDB and then SohoPump. Another dialog asks for the x and y coordinate fields. This is an easy choice in this exercise; X Coord is the X coordinate and Y Coord is the Y coordinate. Click on NAME in the choice in the lower left so that the names of the pumps will be displayed. To make the pumps large, choose 15 for their size, and change the color by clicking in the black rectangle and choosing a brighter color like red. Click OK after choosing the color, and then again for the dialog. You should see the pumps and their names arrayed on the map.

To display the cases, repeat the ADD POINTS process, this time choosing SohoDead from SAMPLE.MDB and 5 for the size of the symbols. Use a contrasting color like black and do not choose a Point Label field for displaying text. You should see a large number of points representing the addresses where cholera cases occurred. Since some households had multiple cases, the visible points actually represent “households having at least one death due to cholera” rather than single cases.

Although the map suggests that the Broad Street pump was central to the location of cases, and other evidence did incriminate this pump, one must remember that the distribution of points is a function not only of the mortality rate but also of population distribution, and that rates rather than numbers of cases would be necessary to draw a scientific conclusion. Assuming, however, that the population was fairly evenly distributed through Soho, the impression that the map gives is useful. Dr. Snow’s similar

map helped him convince the neighborhood council to remove the handle of the Broad Street pump, and the epidemic subsided.

Guided Tour of the VisData Program

Highlights

- Seeing the inside of the language translation database
- Seeing the inside of an Epi Info View table

VisData is a utility program for viewing and performing useful operations on database files or tables. Choose Visualize Data from the main menu. Choose OPEN from the FILE menu and the MICROSOFT ACCESS file type (Epi Info 2000 tables are in the Access 97 format). Open the SAMPLE.MDB database. The buttons on the toolbar control what happens when you click the icon for one of the tables. Tool tips explaining the function of each button appear if you allow the mouse to hover over a button. Click on the leftmost toolbar button and then the one with the grid just to the left of the blue cylinder. Double-click the table name AgeWithCount to see a spreadsheet view of the data in this table. Note that the variables are represented as column headings and records as rows. *VisData* is a useful utility for seeing the internal details of databases. Microsoft Access, if you have it, can be used for the same purpose.

Next Steps

This completes the Guided Tour. You are now familiar with the main features of Epi Info 2000. We suggest that you try working with your own data, either by designing a View and then entering data or by READING an existing dataset in the *Analysis* program. The rest of the manual provides more detail on the programs, and the How To chapter has instructions for solving specific problems. Reference material is in the Commands chapter and is also available from the programs via the Help buttons in particular dialogs.

We hope that you have enjoyed the Tour, and will send your impressions and suggestions to the Development Team at the contact addresses in the front of the manual. If you would like to participate in the Epi Info Worldwide Discussion Group (a LISTSERV) by e-mail, please use the instructions in the front of the manual to join. The Epi Info Web Site is available at all times to provide information, updates of programs, and access to vendors, trainers, and the Epi Info Technical Support staff.

Data Management



Importing, Exporting, Relating, and Accessing Data in Diverse Formats

Microsoft Access Databases

The Microsoft Access database file format is probably the most popular format for Windows databases. An Access database file has the extension .MDB, and is basically a DOS/Windows file residing somewhere on a storage disk. The internal features cannot be seen or edited in a word processor, although the VisData program provided with Epi Info displays the contents and many of the properties of the database and its component tables. A REPAIR feature is provided in VisData for correcting minor problems with corrupt files, but clearly this function has its limitations.

An MDB file contains one or more tables (up to 1000, in theory) that are arranged logically in columns (variables) and rows (records). Because the amount of data stored in text fields, for example, can vary, the physical arrangement of data within the file is more complicated than simple rows and columns.

Epi Info 2000 databases are built around two types of tables within an MDB. The screen appearance of the data, including fields for data entry and check code commands, is contained in a View table, with variables defined for Epi Info use to describe features such as the NAME, PROMPT, and various size and location parameters of each field. Epi Info data tables, however, are simply columns to represent variables in the questionnaire and rows to represent individual records. Data tables are essentially the same as those used in Microsoft Access, although Epi Info, for reasons of compatibility does not use stored procedures or relations in the database tables.

Epi Info 2000 Views

An Epi Info 2000 database in its simplest form consists of a special table called a View and a Data table, both in the same MDB file. Think of the MDB as a project wrapper, in which the various tables of a project can be stored, making it easy to copy the View/Questionnaire, the Data, and even *Analysis* programs associated with the data to another disk location or another computer.

A View table has rows and columns defined in a way unique to Epi Info 2000. It has a row (record) for each field on the screen giving the name of the variable, the prompt, the type of data, locations on the screen of the prompt and the field, and an indirect indication of the data source. Check code, if any, is stored in the View record for each field. Everything necessary for displaying the questionnaire on the screen is contained in the View. Because variables are stored by page, only the variables on a single page are

retrieved from the disk at one time, allowing performance to be maintained for very large questionnaires.

Fields in a View are matched to Variables (columns) in a data table by name. If the name of the variable matches, data items are retrieved from the Data table. If there is no corresponding variable in a data table, one is created, since having a field on the screen (other than a text label) with no place to store the data would create a problem. The reverse is not true; a data table can contain many variables for which no fields are provided in the View, thus allowing for different "Views" of the same data.

Epi Info 2000 Data Tables

After a View is created, the *Enter* program automatically creates a Data table if none is yet specified in the View. After this occurs, the name of the Data table or tables is stored in special SOURCE records in the View. As we will see later, MetaViews consisting of SOURCE records only can be defined so that check code and field definitions can be applied to more than one Data table. A state health department might maintain separate Data tables but only a single view for a number of counties, for example.

Epi Info Data tables are designed to be the simplest kind of Microsoft Access table, without stored procedures, integrity constraints, or defined relations to other tables. All the complexity is confined to the View so that the Data tables can be readily used in Microsoft Access or other software systems if desired.

Microsoft Access Links

The Microsoft Jet Engine, the database engine behind the scenes in Epi Info 2000, provides a mechanism called a Link, for accessing files in more than one MDB. Links function somewhat like the shortcut icons on the Windows Desktop. Just as clicking on a shortcut opens a file that is not on the desktop, referring to a Link table (LNK) has the effect of being able to read and write in a table in another MDB, or in the case, of files like dBASE or Paradox, in an entirely different kind of database.

What are Related Tables, and Why Use Them?

Suppose that you do a survey of households and record for each one some basic information such as address, number of rooms, and head of household. You also interview each family member and have information for between 1 and 15 such persons per household. In thinking about how to capture the data, there are two obvious choices:

Create a separate questionnaire for each person, and copy the household information onto up to 15 forms per household

Or

Use a single form per household and allow up to 15 lines for individual data on the form

If the forms are computerized, the first method requires redundant and error-prone entry of the same data a number of times for a household. The second works well on the written form, but leads to difficulties when I try to analyze data from fields called

PERSON1, PERSON2, PERSON3, etc., even assuming that individuals are entered in some defined order.

One day, while jogging, you might come up with the idea of using a single form for each household, and another form for each individual, and linking the two by using a common Household number. This minimizes the amount of paper needed, and also permits easier analysis, since the forms for each person are in the same format. By using the Household number, I can look up household information for any given person whenever it is needed. Or, by keeping the forms in order, I can simply staple the household form to its group of person forms.

Relational database features in the computer work in very similar fashion. By using unique identifiers, I can "relate" records in one dataset (household) to those in another (person). The computer automatically looks up the necessary information in both records when needed, so that each person appears to have a copy of the correct household information. Hence, I can use TABLES WATER ILL to find out how many ill and well persons (ILL in the Person View) have running water at home (WATER in the Household View).

Creating Related Tables

Creating a related View in *MakeView* requires defining a button on the parent form that will allow access to the related View. To see how this works, run *MakeView*, choose MAKE NEW VIEW from the FILE menu, and then create a new MDB called RELATE.MDB. Within this, give the name of the new View as HOUSE. On the HOUSE View, create text fields for NAME and ADDRESS, and a Yes/No field for Private Water. Now right click on the form to bring up the field dialog and enter CHILDREN as the Prompt. Click on the button called CREATE RELATED VIEW. In the next dialog, choose FORM SHOULD BE ACCESSIBLE : ALL THE TIME and then OK. In the next dialog, choose CREATE RELATED VIEW and click OK.

Now make a second related view button with the Prompt WATER in the same way, but this time the choices are:

```
CREATE RELATED VIEW
FORM SHOULD BE ACCESSIBLE ONLY WHEN CERTAIN CONDITIONS ARE TRUE
RETURN TO THE PARENT FORM WHEN ONE RECORD HAS BEEN ENTERED
In the box for CONDITIONS, enter PRIVATEWATER = (+)
```

You should now have buttons called CHILDREN and PRIVATE WATER. Let the mouse cursor hover over the CHILDREN button and note the instructions. Hold down the Ctrl key and left click on the button. You should see a blank view called CHILDREN. Create a text field for NAME and a numeric field for AGE, adding any others that you think relevant to record the status of a child in the household. Click on the BACK button to return to the main form.

Now do the same thing with the PRIVATEWATER button, creating a View with questions like WATER SOURCE and DATE OF LAST COLIFORM TEST. Click on BACK to return to the HOUSE View. From the FILE menu, choose ENTER DATA.

Entering Data in Related Tables

In the HOUSE view enter a name and address. In PRIVATE WATER, choose NO and note that the WATER button remains inactive. Click on CHILDREN and then enter data for one or more children, clicking on NEW for each new child's record. Click on BACK to return to the HOUSE View. Click NEW to save the record and enter another name and address. This time answer YES to PRIVATE WATER and the WATER button should become active. Click the WATER button and fill in the form for WATER. Click NEW and note that it has the same effect as BACK, since only one instance of the WATER form per household is allowed. Click NEW to save the second HOUSE record and then choose EXIT from the FILE menu. You have created a View called HOUSE and two related Views called CHILDREN and WATER. CHILDREN is unconditionally related to HOUSE and WATER is only accessible if PRIVATE WATER has the value YES.

Analyzing Related Tables

A View with related Views can be read in *Analysis* using the READ and RELATE commands. If the database was created in *Makeview* and the data entered in *Enter*, *Analysis* will establish relationships automatically, using the UniqueKey and Fkey (Foreign Key) variables created by Epi Info 2000. If the relationship was created in Epi Info 6 or another program with that uses different keys, the keys variables in both files must be identified. A relationship in Epi2000 is represented by “:.” (Without the quotes).

```
Read 'D:\Epi2000\sample.mdb':file1
Relate 'D:\Epi2000\Sample.mdb':file2 Key1::Key2
```

Analysis can establish relationships using multiple keys. For example:

File A contains SSN and Date while file B contains SSN1 and Date1

To relate files that match both criteria at the same time (same SSN and same Date) the relational key might be:

```
Relate 'C:\epi2000\sample.mdb': viewRelated SSN::SSN1 and Date::Date1
```

Epi Info 2000 comes with a related database called REFUGEE.MDB. The structure for the system is shown below. Since Refugee was imported from Epi6, all relationships must be declared in *Analysis*. Household (Family) is related to Person (Patient) using FAMIDNUM key which can be located in both databases. Patient relates with all other tables by BOH, in some cases the relationship is one to one while in another the relationship is one to many.

The following program will read both databases, establish the relationship and analyze data from variables located in both files.

To display the number of refugees arriving from “Anycounty” by sex type the following instructions:

```
READ 'C:\EPI2000:REFUGEE.MDB':viewPatient
RELATE viewFamily FAMIDNUM::FAMIDNUM
```

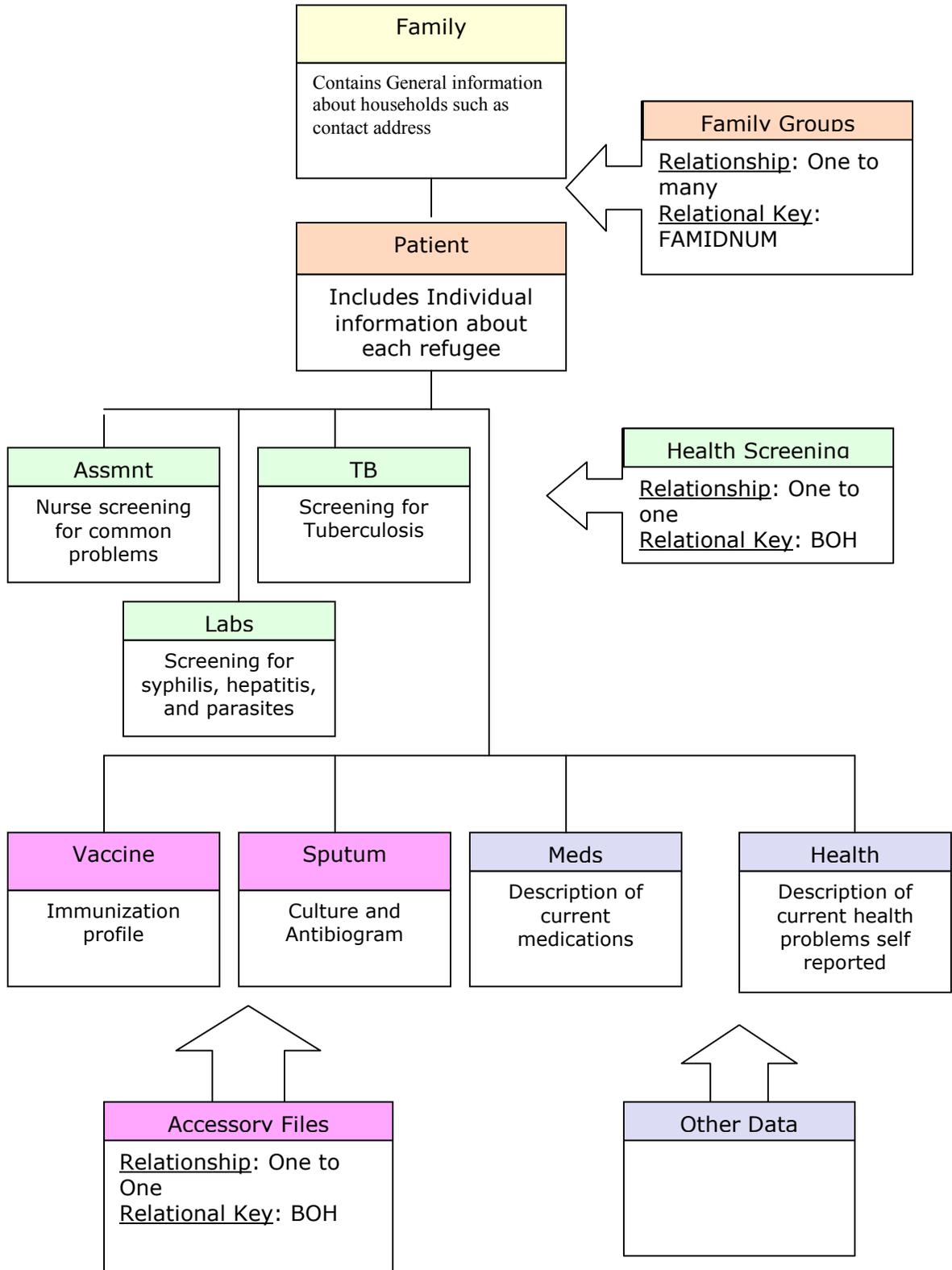
TABLES COUNTRY SEX

The variable for COUNTRY of origin is located in the Family table while SEX is in the PATIENT view. The first instruction READs the first table (Patient), the second one (RELATE) establishes a relationship between table1 (Patient) and Table2 (Family). After the relationship is successfully created, variables in both tables are available for use in the table.

The keys for relating can also contain mathematical or string concatenation expressions. Suppose that two tables contain related records but that someone has added 1000 to each of the keys in one of the tables. The relationship can be established for *Analysis* by using (say) KEY1+1000 as one of the keys and KEY2 as the other.

You can use any of the Epi2000 functions to determine the relationship, although this is rarely necessary, and it may be easier to write out a new file in which the complex key is included as a single variable, and to use this single variable as a key.

REFUGEE.MDB



Analyzing Data in Non Epi Info Files

To analyze data in a file or dataset such as Paradox, FoxPro, dBASE, or other format, a current project must be defined. The current project is shown in a gray box at the top of the READ dialog. To READ a database in any format but MDB, choose the desired data type in the READ command dialog. Files with the proper extension should be displayed. Select the file you want to open, and click Ok to confirm it. Depending on the database format you may be asked for additional information; for example if the data source is an Excel table, you will be asked which sheet to open (only one page can be opened at the time). In other cases such as text files you need to select whether the data is stored as fixed length or delimited and perhaps specify variables names, types and sizes, or delimiters. If the database is in dBASE format, no further questions will be asked.

When *Analysis* has all the information required to read, it will create a link to the table selected. In some cases you may want to create a permanent link (like a short cut so that you do not have to fill all these information again). *Analysis* will ask for a name for that link. If no name is provided, the link will be called lnk# where # represents a sequential number. Those links that are named lnk# will be deleted when the project is changed or *Analysis* is closed whatever occurs first. If you rename a link to lnk#, *Analysis* will assume that is a temporary link and it will delete it at the end of the session.

Once created, a permanent link can be opened just like a regular table. To see available links in the read form is necessary to select "All" in the SHOW frame.

In *Analysis*, it is possible to edit data in a database stored in another format (such as dBASE) by using the LIST command and selecting "Allow Updates". The *Enter* program, however, does not write to file formats other than Microsoft Access/Epi2000.

ODBC File Access: Setting Up a DSN Link

Some types of data sources, those accessed through the ODBC or OLE DB methods, require that a special file containing access information be constructed and stored before they are READ in *Analysis*. To construct a DSN (Data Source Name) link to an ODBC or OLE DB database such as Microsoft SQL Server, Oracle, Sybase, or Informix, choose SETTINGS from the Windows START menu. Within the Control Panel that appears, click on the icon for "ODBC Data Sources (32bit)". On the tab called FILE DSN click the ADD button, and then from Create New Data Source, choose the driver to be used. Now click NEXT and use the browse button to navigate to the directory where Epi Info 2000 is installed, probably ..\EPI2000. Enter as the Data Source Name, "Epi 2K Sample Database." Click SAVE. When you are prompted for More Information, click the SELECT button under Database and find the SAMPLE.MDB file in the Epi2000 directory. Now click OK in the ODBC Microsoft Access Setup window and you should return to the User DSN display. Allow Windows to place the file in the

specially designated directory for DSNs. Click OK again and then close the Control Panel window.

Database Analysis Using ODBC

The challenging part of setting up an ODBC file connection is in creating the DSN (Data Source Name) as described above. Once this is done, use the READ command in *Analysis* and select ODBC as the Data Type. You should see the available DSNs or be able to navigate to the directory where you placed yours. When you use a DSN, there may be some security formalities, such as entering a username and password, or these may have been included when the DSN was constructed, and hence will not appear again. Although there may be restrictions on the type of access provided to particular tables or fields, analysis of an ODBC data source should otherwise be similar to working with a native Epi Info 2000/Access dataset.

Importing Epi Info 6 Files

Analysis imports Epi Info 6 files automatically into a temporary Microsoft Access table. It is also possible to import data from Epi6 into the MDB format by choosing IMPORT from the FILE menu in *MakeView*. After supplying name and location of the file, a new Epi Info 2000 View and Data table will be created automatically. The fields in the View may require some adjustment for best appearance in the Windows screen, and you may choose to enhance the View by adding Groups and other features. After analysis is completed you can export your MDB into Epi6 format using the write command.

Importing Epi Info 6 Check Code

Check code in Epi Info 2000 is not quite the same as that in Epi 6 check (.CHK). It is close enough, however, so that many features can be reused. A Check (.CHK) file from Epi Info for DOS can be transferred to appropriate variable blocks of Epi Info 2000 check code by running IMPORT EPI 6 CHK FILE on the *MakeView* FILE menu. Each line begins with an asterisk so that the code will be ignored until you make the necessary changes in syntax, remove the asterisk, and test the code in Epi Info 2000.

Importing a Related File System from Epi Info 6

The IMPORT function in *MakeView* imports one file at a time from the Epi Info 6 format. In Epi Info 6, related files and keys for the relations are designated within check code in the .CHK file.

To import a system of related files from Epi 6, first make a diagram on paper of the system to understand which file is the main file, which files are related and the keys for each. Consulting the check code and the Epi Info 6 users manual may help with this step. If there are IF statements to express conditional relationships (IF <fieldname> = something or other THEN RELATE <Filename>), then jot down the conditions.

Make a proper backup copy of all the files you will be converting, and set it aside. Now import each of the files into Epi Info 2000 using the *MakeView* FILE|IMPORT menu choice. On the main View, right-click in a suitable location to create a relate button, and

follow the procedure given above in the section on Creating a Related View. Instead of choosing to CREATE A NEW RELATED FILE, choose RELATE TO EXISTING VIEW, and make appropriate choices to create the relationship to the existing View that was imported from Epi Info 6.

The relational keys will be the same in both Views, since this is enforced by Epi 6. To enter data in Epi2000 is necessary however to convert the relational system used in Epi Info 6 to the new format in Epi2000. When a relation button is created in *MakeView*, a new field called Fkey (foreign key) is added to the related table. To convert an Epi 6 relationship into an Epi2000 relation is necessary to fill the fkey field with the appropriate data. This can be done in *Analysis*. The MERGE command can move values from the unique key of the parent table to the fkey of the child table when the word “Relate” (Without quotes) is added to a valid syntax. Since this is not a common procedure (it will be use one time per system) there are no boxes in the merge dialog to click on.

Read the child table, then select MERGE from the command list. Supply the appropriate information in the MERGE dialog including the old relationship (the relationship used by Epi 6). When ready instead of using the OK button, click on “Save Only”. *Analysis* will write the appropriate syntax in the program editor. Identify the merge line and at the end of the statement add the word “RELATE” (without quotes). Click on run this command only.

Once these steps have been completed, test the new relationship in *Enter* by choosing ENTER DATA from the FILE menu. Continue this process until all the relationships have been reestablished in the Epi Info 2000 Views.

This procedure has to be completed every time a new set of related tables in Epi 6 is converted to Epi 2000. Therefore you must decide whether to use Epi 6 or Epi 2000 for data entry. It is not possible to use the same dataset in both systems, once it is converted to Epi Info 2000. Do not attempt to MERGE an Epi Info 6 file with an Epi Info 2000 dataset, as data may be lost or misidentified.

Analysis (PGM) Programs?

Programs written in *Analysis* can be stored in the current MDB. This is done automatically in a separate table called “Programs”. Do not create another table or view called ”Programs” because *Analysis* maintains its own version. *Analysis* will store some information about the program such as comments, author, and date of creation.

In some cases it is necessary to share a program between users without sharing data tables (MDB), or to send a program elsewhere, or create one in a Word processor or from DOS batch file commands in the menu. *Analysis* can read .PGM files in text format containing program commands. To read a program stored in a .PGM file, click on the “Open” button located at the top of the Program editor. In the Open form click on the Text File button. A window to select the text file will be displayed. Choose a file and click Ok to load it into the Program editor. In general simple Epi 6 programs can be executed, but those that point to different files and folders will need some editing to use

the Epi Info 2000 syntax.. Consult the COMMANDS and FUNCTIONS chapters for details on syntax.

HTML Output Files

Output files can be placed in any folder you want. The ROUTEOUT command will allow you to select both path and filename. If no output file is selected, *Analysis* will use the default value (see below). In every folder *Analysis* will create a new index table that contains links to the files created.

The default name and path for HTML documents can be defined using the STORING OUTPUT command. The first part of an output file name must begin with a letter and can contain any valid character. The second part of the name contains the the output file sequence which is a number. The default values for automatic output are “OUT” as prefix and 1 as sequence number. If you change the output prefix to “page” and set the sequence to 100, the first output created for analysis will be called “Page100.htm.” It is best to keep filenames within the standard DOS 8+3 format, since some Local Area Networks and other programs still do not accept the long file names that are convenient for most purposes in Windows.

The archive database allows renaming and keeping important results permanently, since output names are recycled after the time or number of files designated in the OUTPUT STORAGE dialog.. The archive database will be stored in the same directory in which the INI file is located. Unlike the results index, the archive can contain information about files located in different folders.

Many Views for A Data Table

In some cases is necessary to have different views from one table. For example, you may not want to show confidential information to your secretary, or clinicians and epidemiologists in a clinic need access to different kinds of information in the same file.

Epi2000 can solve this problem using a single database by having more than one View connected with the data table(s). One View can contain all fields and others can contain only a subset.

A View cannot contain fields that are not in the dataset, as there would be no place to store data that might be entered. If such a View is created, the necessary fields will be added to the database automatically to prevent problems.

To create a new view for an existing table, use the COPY VIEW option in VIEW menu in *MakeView*. Choose CREATE A NEW VIEW USING THE SAME TABLE. Create a second view by choosing the MDB in which to place place the View, give the View a name, and click Ok.

Backing Up Data Files

An essential feature of data management is maintaining backup copies of databases so that most of the data can be restored in the event of hard disk failure, computer virus attack, fire, flood, or power surges. In all but trivial applications (defined as such by their lack of backup), multiple rotating backup copies and off-site storage are worth the extra

trouble. Several generations of backup copies will help protect against the virus that your computer acquired last week that did not show up until this week, and against the common human tendency to nervously destroy the first backup copy when a crisis occurs.

A variety of media and compression techniques are available for backup. MDB files compress very well, sometimes to 5% of their original size, using ZIP or other compression methods. A copy of the ARJ file compression program is provided with Epi Info, and this can be used to compress and back up data from a menu entry with a little programming.

Read-only CDROM copies (CD-R) copies are durable and well protected from overwriting. Whatever medium is chosen should be tested periodically by actually restoring and testing databases, as even commercial backup programs occasionally provide cheerful assurance that all is well, but fail to deliver the data when challenged. Restoration drills are particularly important if you depend on backups made by others—perhaps a LAN administrator—but untested in practice until too late. Systems based on human memory and behavior may fall behind unless supplemented by automatic reminders or automated backup systems.

How To...



Specific Tasks and Frequently-Asked Questions

Overview

This chapter contains answers to specific questions that may arise about performing tasks. Additional information in the popular format of "Frequently Asked Questions (FAQ)" may be found on the Epi Info website.

EPI 2000 Menu

Develop a new menu (.MNU) file (*Epi 2000 Menu*)

To develop a new menu file, examine EPI2000.MNU in the Word Processor. Make your own .MNU file, using SAVE AS on the FILE menu of the Word Processor. Save a copy of EPI2000.MNU by another name, and edit the menu items and commands to your own taste. (See MENU in the *Commands* chapter.)

Change the picture on the menu (*Epi 2000 Menu*)

Select SETTINGS on the menu, then click on PICTURE. From the file dialog that appears, select a suitable image file of the .BMP type. Other types of images can be converted to the .BMP format by using commercial or shareware image editing programs.

Images can also be specified on the command line when running the menu program from the DOS command line, from the RUN command on the Windows START menu, or from another program.

```
C:\Epi2000\EPI2000 EPI2000.MNU REFCAMP.BMP
```

This will display the Epi2000 menu, but with the picture normally shown on the Surveillance menu example instead of the EPI2000.BMP image that is specified in the EPIINFO.INI file.

A third method of specifying an image is with the PICTURE command.

```
PICTURE REFCAMP.BMP
```

Inserting this command in a block of commands in a .MNU file allows a temporary change of the menu image. (See PICTURE in the *Commands* chapter.)

Add or change a menu item (*Epi 2000 Menu*)

To add or change a menu item, edit the .MNU file. For example, to add an item to run Microsoft Word, first add the MENUITEM command to the .MNU file. Enter the words that will appear on the menu, followed by the command block name. An ampersand (&)

can be used to choose a single character “hot key.” Hot keys must be different for each item on a simple menu.

```
MENUITEM "&MS Word", DoWord
```

Now add a CommandBlock section:

```
DoWord
```

```
Begin
```

```
EXECUTE.EXE WINWORD.EXE
```

```
End
```

A menu entry is created on the POPUP menu within which the MENUITEM occurs. When the item is selected, the commands in the CommandBlock named are executed. (See MENUITEM in the *Commands* chapter.)

Add or change a button (*Epi 2000 Menu*)

To add or change a button, edit the .MNU file. The BUTTON command is used to create a button. The command

```
BUTTON "&NOTEPAD", DoNotepad, 50,80
```

creates a button on the screen with the caption “Notepad” and the locations given as percentages of screen width and height: half way across the screen (50%) and (80%) of the down from the top. Clicking on the button with the left mouse button activates the commands in a block called DoNotepad. (See BUTTON in the *Commands* chapter.)

Program the response to a menu item or button (*Epi 2000 Menu*)

A named block of commands is needed in the .MNU file for the program to respond to a menu item or button.

For example, the BUTTON “MakeView,” when selected by the user, runs the commands in the block called MakeView, which looks like this in the file called Epi2000.MNU:

```
MakeView
```

```
Begin
```

```
EXECUTE MAKEVIEW.EXE
```

```
End
```

You can create these lines or edit the ones that are already there. A number of menu or DOS batch file commands can be inserted between the Begin and End markers. (See MENUITEM and BUTTON in the *Commands* chapter.)

Set the working directory (*Epi 2000 Menu*)

Select SETTINGS on the menu and click SET WORKING DIRECTORY.

Turn buttons on or off (*Epi 2000 Menu*)

Select SETTINGS on the menu and click BUTTONS ON OR OFF.

Let users set the working directory from a menu (*Epi 2000 Menu*)

Insert the SETWORKDIR command into a block of commands in a .MNU file. For example

```
WorkingDir
Begin
SETWORKDIR "Please choose a working directory for which you have write
privileges."
End
```

The menu changes the logged directory to the specified directory before executing commands. (See SETWORKDIR in *Commands* chapter.)

Let users turn buttons on or off within a menu (*Epi 2000 Menu*)

Insert the SETBUTTONS command into a block of commands in a .MNU file, as in:

```
ButtonSetting
Begin
SETBUTTONS
End
```

(See SETBUTTONS in *Commands* chapter.)

Send information to other programs using permanent variables (*Epi 2000 Menu*)

Use the ASSIGN command to create a variable and assign a value, as in:

```
ASSIGN CurrentYear= "2000"
```

The value of LANGUAGE becomes available as a "permanent" variable within *Analysis* and in check code written in *MakeView*. In menu (.MNU) programs it is not necessary to define variables; they are defined automatically by being included in an ASSIGN statement. (See ASSIGN in *Commands* chapter.)

Bring up the help system (*Epi 2000 Menu*)

Epi Info 2000 provides help on a variety of topics. To bring up the HELP files you can go to MANUAL on the menu and click on the topic with which you need help. The Main Menu also provides a button to allow easy access to the Guided Tour, which can help you get acquainted with the programs.

MAKEVIEW**Make a view (*MakeView*)**

Open *MakeView* by clicking on the MAKEVIEW button on the Main Menu screen or by going to the PROGRAMS menu and selecting MakeView (Questionnaire). Right click under Page Names and select MAKE NEW VIEW. You can also select MAKE NEW VIEW from the FILE menu.

Insert a grid (table) in a view (*MakeView*)

Right click on the page to open the Field dialog. Enter a prompt and click on the INSERT GRID button. Enter the name of the columns one at a time; click on SAVE COLUMN after each entry. When finished, click DONE at the bottom of the dialog.

You can resize the grid by clicking on its prompt and dragging the handles that appear while holding down the left mouse button. Columns can be resized by placing the mouse over the line between columns until a horizontal arrow appears. Then hold down the left mouse button and resize the column.

Line up fields (*MakeView*)

Select fields to be aligned by holding down the left mouse button and dragging to draw a dotted line around the fields. Select FIELDS from the menu, click ALIGNMENT, and select VERTICAL or HORIZONTAL.

Insert a background picture or color on a page (*MakeView*)

Select VIEW from the menu and click BACKGROUND. When the **Background for Page** dialog comes up, choose whether you want a background color or an image. Click in box of desired action. Once you click in a check box, either a **Color Scheme** dialog or **Background Image** dialog will come up. Choose desired color or image and click OK.

Group fields together as a functional unit (*MakeView*)

Select fields and the area for a group by holding down the left mouse button and dragging to draw a dotted line around the fields. Select FIELDS from the menu, then select NEWGROUP. When the **Group** dialog appears, type in a group description or name, choose the desired color, and click OK.

Cut, Copy, and Paste fields (*MakeView*)

Select fields to be cut or copied by dragging a rectangle around them with the left mouse button down, go to EDIT, and choose which action will be taken. To paste, go to EDIT and click on PASTE.

Delete an existing data table without deleting the view (*MakeView*)

Choose DELETE EXISTING DATA from the VIEW menu and respond to the warnings and questions that appear.

Change the Prompt, Name, or Type of a Field in a view (*MakeView*)

Right click on the field that needs to be changed. When the **Field** dialog appears, make the changes needed and click OK.

Create a Related Form/View (*MakeView*)

Right click on the screen to open the **Field** dialog. Enter a prompt and click the button called CREATE RELATED VIEW. When the **Conditions for Related Form to Be**

Active dialog appears, select ANYTIME from the options and click OK. The next dialog is the **Related View Choice** dialog, where you can choose to create a new related view or relate to existing view. By choosing to create a new related view, you are creating a new view; this allows you to get another screen on which to create more or different fields. By choosing to relate to existing view, you are referring to another view or another form that already exists.

Move Related Field buttons (*MakeView*)

Hold down the SHIFT key and the left mouse button and drag the field to the desired location.

Convert a file from Epi Info, Version 6, format to Epi Info 2000 (*MakeView*)

Select FILE from the menu and IMPORT. When the **Select Import File** dialog comes up, select the .REC file from Epi Info, Version 6, to be imported. Click OPEN and choose a destination file (.MDB) in Epi Info 2000. After choosing the MDB, accept or edit the name of the table to be created within the MDB.

Make a view of an existing database (*MakeView*)

Select FILE on menu, click on MAKE VIEW, and select MAKE NEW VIEW FROM EXISTING DATATABLE.

Compare two dates to obtain an age or duration (*MakeView*)

An example is found under *Checking and Controlling Data Entry* in the Guided Tour of the *MakeView* program.)

Specify Legal Values (*MakeView*)

When creating a new field, you can choose legal values for that field by clicking the LEGAL VALUES button in the **Field** dialog. Choose CREATE NEW, and enter suitable values in the list that appears, pressing Enter after each to move to a new blank line. Click OK, then OK again in the **Field** dialog.

Do Skip Patterns (*MakeView*)

Skip patterns can be created by changing the FIELD ORDER or by using the GOTO command in the PROGRAMS section. FIELD ORDER (TABORDER) controls the sequence in which the cursor visits fields on a questionnaire. To change the FIELD ORDER, go to FIELDS on the menu and select FIELD ORDER. In the FIELD ORDER dialog, use the UP and DOWN buttons to move a selected field up or down in the entry order list and thereby change the order of the fields.

To use the GOTO command, click on PROGRAMS on the Page Name Panel. Click on the arrow under CHOOSE FIELD WHERE ACTION WILL OCCUR to pull down a list of fields. You can use the GOTO command under the FIELDS tab, or you can type in a GOTO program using IF statements in the Program Editor.

Do automatic coding between fields or into the same field (*MakeView*)

Open the **Field** dialog, enter your prompt, and click CODES. Select the field(s) to receive codes. Existing tables can be used, or a new table can be created to set up code fields.

Set up conditional related forms (*MakeView*)

Right click on the screen to open the **Field** dialog. Enter a prompt and click the button called CREATE RELATED VIEW. When the **Conditions for Related Form to Be Active** dialog appears, select ONLY WHEN CERTAIN CONDITIONS ARE TRUE from the options. When the **Access** section appears, choose which variables or values are to be true before accessing this page, then click OK. The next dialog is **Related View Choice**, where you can choose to create a new related view or relate to the existing view. By choosing to create a new related view, you are creating a new view, which allows you another screen to create more or different fields. By choosing to relate to existing view, you are referring to another view or another form that already exists.

Change the Prompt or Type of a Field in a View (*MakeView*)

If *MakeView* has not yet made a data table, or there are no records in the data table, names, field types, and patterns can be changed by right-clicking on the screen prompt for the field. When the field dialog appears, change the desired characteristic, and click OK. Then Save the view, either by choosing SAVE from the FILE menu or in response to a reminder. If a data table exists and contains data, you can change the *prompt* by popping up the field dialog in *MakeView* as above. Once the data table contains entries, you cannot change the field *name*, but the field *type* can be changed. *MakeView* will try to transfer the data into the new type, but in some cases must discard incompatible data items. Changing the type of a text field to numeric will transfer numeric data, but cannot deal with a number like "M0111," and will assign a missing value because of the "M" character it contains. A text field can safely be changed to a multiline field, however, since both contain text.

ENTER

Change the values displayed in Yes/No fields (*Enter*)

Click on OPTIONS on the menu.

Move from record to record (*Enter*)

Move to the first or last records by using the double arrows in the lower left panel of the **Page Name** section. By using the single arrows, you can move one record at a time. Clicking the NEW button brings up the next available empty record. The F7 and F8 function keys also move to the previous or next record, as in Epi Info 6.

Save a page (*Enter*)

Pages are saved automatically. The current page is saved when you move to the next record or page.

Save a record (*Enter*)

Saving records can be done by clicking on SAVE DATA on the **Page Names** panel or by clicking on NEW. Records are also saved automatically if you move to another record.

Move among parent and related forms (*Enter*)

The HOME and BACK buttons can be used to move to and from parent and related forms. These buttons do not appear until you have displayed a related View. The HOME button will take you back to the main form. The BACK button displays the parent of the current form. In a two-level, hierarchy, both buttons accomplish the same thing, but with three levels, HOME skips back to the main form and BACK moves up just one level.

ANALYSIS**Read Dbase, Paradox, FoxPro, Excel, and Microsoft Access database files (*Analysis*)**

Click on READ on the **Analysis Commands** panel. From the **Project Format** dialog, choose the file you want to read. If it is an Access/Epi2000 file, view the tables within the .MDB file. (See READ in *Commands* chapter.)

Read an ODBC database such as SQL Server or Oracle_ (*Analysis*)

This is described in the Chapter on Data Management.

Read from one file format and write to another_ (*Analysis*)

Click on READ from the **Analysis Commands** panel. Choose the correct format for your file. After reading the file, click on the WRITE command, and from the list of output formats (file types), choose the one to write to. Give your file a name, choose the desired directory, and click on SAVE and then OK. (See WRITE in the *Commands* chapter.)

Select records_ (*Analysis*)

To select particular records based on criteria that you specify, click on SELECT from the **Analysis Commands** panel. Enter the condition that you want to specify, click OK, and choose the LIST command to show only those records that meet that condition. (An example is found under *SELECT Command* in the Guided Tour of the *Analysis* program.)

Cancel a previous selection_ (*Analysis*)

Click on the CANCEL SELECT command from the **Analysis Commands** panel. (See SELECT in the *Commands* chapter.)

Process data from a questionnaire study (Analysis)

Click on FREQUENCIES from the **Analysis Commands** panel. In the **FREQ** dialog that appears, select one or more variables, or * for all variables, and click OK. FREQ produces a table from the table(s) specified in the last READ statement, showing how many records have each value of the variable. Records may be included or excluded from the count by using SELECT statements. Those marked as deleted in *Enter* will be handled according to the current setting for SET PROCESS. If more than one variable name is given, FREQ will make a separate table for each variable. (See FREQ in *Commands* chapter.)

Combine data in more than one table (Analysis)

Click on RELATE from the **Analysis Commands** panel. To use RELATE, a dataset must have been made active with the READ command. The table to be linked must have a key field (such as HOUSID) that has identifiers relating the records in the two tables. After issuing the RELATE command, the variables in the related table may be used as though they were part of the main table. (See RELATE in *Commands* chapter.)

Process only certain columns (variables) (Analysis)

Click on LIST from the **Analysis Commands** panel. In the **LIST** dialog that appears, choose one or more variables to be listed. (See LIST in *Commands* chapter.)

Obtain an Odds Ratio, Relative Risk, Chi Square and Fisher Exact test (Analysis)

These are provided by the TABLES command, described in the Guided Tour and also in the *Commands* chapter.

Obtain exact statistics (Analysis)

Exact statistics are provided by the TABLES command, described in the Guided Tour and also in the *Commands* chapter. Further details are provided in the *Statistics* chapter.

Do an ANOVA test (Analysis)

Click on MEANS from the **Analysis Commands** panel. Choose the name of a numeric variable containing data to be analyzed and the name of the variable that indicates how groups will be distinguished. Depending on the size of the table and whether the data are normally distributed, MEANS provides the equivalent of ANOVA for two or more samples. (See MEANS in *Commands* chapter.)

Do a Student's t-Test (Analysis)

Click on MEANS from the **Analysis Commands** panel. Choose the name of a numeric variable containing data to be analyzed and the name of the variable that indicates how groups will be distinguished. Depending on the size of the table and whether the data are normally distributed, MEANS provides the equivalent of STUDENT'S T TEST for two samples. (See MEANS in *Commands* chapter.)

Add population data to a dataset (*Analysis*)

Create a related view with a variable containing geographic names (e.g., county names), and one or more variables containing population numbers. Use the RELATE command to join this table to summary data containing the same geographic names. Rates can be calculated by DEFINEing a new variable and ASSIGNing it the value of a numerator variable divided by the population variable.

Change the appearance of a graph (*Analysis*)

Click on GRAPH from the **Analysis Commands** panel and use the GRAPH dialog to display a graph. To customize the graph, click on MENU, select TOOLS, then TOOL BAR. When the toolbar appears, click on the sixth icon, the **Change Gallery Type** icon. Experiment with the other icons and with the color palette that can also be displayed from the TOOLS menu. (See GRAPH in *Commands* chapter.)

NutStat**Open or create a data file (*NutStat*)**

To open a file, run *NutStat* from the Epi Info 2000 menu. When *NutStat* appears, choose the FILE menu and then OPEN. To create a new file, click FILE on the menu and select NEW. When creating a new file, place a check mark in the **New File** dialog at bottom of the **Select a Table** dialog.

Set up units of measurement (*NutStat*)

From the VIEW menu, choose CUSTOMIZE. From the **Customize Anthropometric Data** dialog, click the UNITS tab. Choose international or US reference standards

Assess height and weight (*NutStat*)

Enter the height and weight on the main screen of *NutStat*. Click RECUMBENT if the child was measured lying down. Statistics will appear on the right side of the screen, comparing the child with others of the same sex and age in the growth reference curves selected in the Customize page.

Assess arm circumference (*NutStat*)

Enter the arm circumference on the main screen of *NutStat*.

Use Nutstat in combination with another data entry view (*NutStat*)

Setting up a button on a View so that *NutStat* can be used in conjunction with the View is described in the chapter on *NutStat*. The View might contain information for a general pediatric visit, for example, and *NutStat* would be used to enter growth measurements.

EPI MAP

Display a map (*Epi Map*)

A complete map can be saved or opened for display using the SAVE or OPEN entries on the FILE menu of *Epi Map*.

Choose a boundary layer (*Epi Map*)

Select FILE from the menu, then MAP MANAGER. In the **Map Manager** dialog, click ADD MAP LAYER from the LAYERS tab and select a Shapefile. (An example of this is found in the Guided Tour of the *Epi Map* program.)

Prepare a data file for use with a map (*Epi Map*)

The .dbf file that is part of the Shapefile (Shapefiles are really 3 or more files-- .dbf, .shp, and .shx) contains geographic names. Be sure that the Shapefile contains polygons and not roads or points. Then READ the .dbf file in *Analysis*, DEFINE a new variable, ASSIGN it the value 0 (zero), and WRITE the new file in the Microsoft Access format. Then you can add data to the new variable in the *Analysis* LIST command with UPDATE selected, or use *MakeView* to create a View from the Data table and enter data in *Enter*. Obtaining the geographic names directly from the Shapefile will assure that they match those in the Shapefile exactly in spelling and format.

Display values from a data file in a map using *Analysis* (*Epi Map*)

Be sure that the data table or file contains a geographic variable with values matching the names of polygons in a corresponding Shapefile. Then use the MAP command in *Analysis* to choose the variables containing data and geographic names in the data file or table, and the field in the Shapefile that contains the same geographic names. After the map is displayed, you can customize it from within *Epi Map* and save the result either as a bitmap (BMP or GIF) or as a MAP, from the FILE menu.

Display values from a data file using *Epi Map* (*Epi Map*)

Run *Epi Map* from the main menu.

Change colors or patterns (*Epi Map*)

Click the GENERAL tab in MAP MANAGER to set a background color.

Move the Legend (*Epi Map*)

Hold down the left mouse button with the cursor over the legend, drag it to a new location, and release the mouse button.

Display Rates (*Epi Map*)

In the MAP command in *Analysis*, choose both a data variable and a denominator variable.

VISDATA

See data in spreadsheet format (*VisData*)

Choose VISUALIZE DATA on the main Epi Info 2000 menu. On the FILE menu, choose OPEN DATABASE and then MICROSOFT ACCESS or another format that matches your database. A panel on the left side of the window should display the tables within the database. Click on the leftmost icon of the toolbar and also on the toolbar icon resembling a grid. Now double-click on one of the small grid icons in the tree of tables. The data in the table will be displayed in row and column format.

Delete or rename a table, field, or file (*VisData*)

Choose VISUALIZE DATA on the main Epi Info 2000 menu. On the FILE menu, choose OPEN DATABASE and then MICROSOFT ACCESS or another format that matches your database. A panel on the left side of the window should display the tables within the database. Fields within the table can be displayed by clicking on the "+" signs next to the icons. To delete or rename a field or table, right click on its icon. On the popup menu that appears choose RENAME or DELETE.

Compact a database (*VisData*)

Click FILE from the menu and choose COMPACT MDB. Select the type of file to be compacted and choose a file.

Repair a damaged database (*VisData*)

Click FILE from the menu, choose REPAIR MDB, and select the files to be repaired.

Samples



Examples Included With the System

Oswego: An Outbreak Investigation

Found in: SAMPLE.MDB:viewOSWEGO

Refugee: A Group of Related Data Views for Families, Individuals, Evaluations, and Specific Conditions

Found in: REFUGEE.MDB

Surveillance: A Menu and Series of Data Input Forms

Found in: SURVEIL.MNU and SAMPLE.MDB:viewSURVEILLANCE

Statistical Examples

Found in: SAMPLE.MDB

A program called pgmSTATISTICS contains all the commands needed to produce the statistical output shown in the chapter on Statistics. To use the program, run *Analysis* and OPEN first the SAMPLE.MDB database and then the pgmSTATISTICS program from the program editor. RUN the program to obtain the output in the Statistics Chapter.

NutStat



A Nutritional Anthropometry Program

Nutstat is a program for recording and evaluating measurements of length, stature, weight, and head and arm circumference for children and adolescents. It can be run as a standalone program or linked to an Epi Info 2000 questionnaire view.

The program calculates percentiles, numbers of standard deviations from the mean (z-scores), and in some cases, percent of median, using data from the following sources:

- The CDC/WHO 1977/1985 reference curves for age, sex, height, and weight (1)
- WHO reference data for arm circumference (8)
- (When available) The 2000 U.S. National Center for Health Statistics reference curves for age, sex, height, weight, head circumference, and body mass index. The release date for the new NCHS data has not yet been announced, and the program does not yet include the data.

The main screen of *NutStat* is configurable to use either English or Metric units and to perform calculations with the desired reference data set. Measurements entered on the screen are automatically stored in a Microsoft Access database and can be retrieved by patient name or identification number. Data from one person or from the entire population can be displayed graphically in several formats. *NutStat* databases can be read by the *Analysis* program to produce summary tables, lists, frequencies, and graphs, or to perform more complex data manipulations.

Existing data in Microsoft Access files can be imported to produce a standard *NutStat* database with appropriate statistics. An Add-Statistics feature adds the results of calculations to external Microsoft Access data files that are not necessarily in the *NutStat* format.

Entering or Editing Measurements in Nutstat

The main screen for *Nutstat* is divided into three general sections, separated by frames. The outer frame contains the ID level data, corresponding to the records for a single person. Clicking the navigation buttons at this level will navigate through the IDs (persons) in the table.

The DATE OF MEASUREMENT frame displays the data associated with a set of measurements on a single date. A person can have many measurements on different

dates. The navigation buttons in this frame will navigate to the other measurement dates for the current ID. The RESULTS frame contained in the DATE OF MEASUREMENT frame contains the calculated statistics for the current measurements. The statistics are calculated upon exit from the relevant data fields.

The GRAPH button will display a graph when clicked. The format of the graph is determined by the selections made in the GRAPH tab of the Customize screen. The graph options can also be displayed upon clicking the GRAPH button if that option has been selected in the Customize screen.

The GRID button will display all of the measurements for a single person in grid format. The values can be viewed but not edited in the grid.

The FIND button provides methods of searching for records based on ID Number, First Name, Last Name, Birth Date, Measurement Date or any combination thereof. The ID Number, First Name and Last Name fields will accept Access wild card characters. The results of a search are displayed in a grid. Double clicking on a record in the grid will close the Find screen and load the record into the Main screen.

The DELETE button will delete the current measurement date and data. If the measurement date is the only one associated with the current ID Number, the ID Number and data will be deleted as well. NOTE: The records are actually deleted from the table and not just marked for deletion, as in other Epi Info programs.

Configuring Nutstat

The configuration screen for *Nutstat* is accessed through the VIEW-CUSTOMIZE menu item. The configuration screen consists of three sections separated by tabs.

The tab labeled DISPLAY contains the names of nearly all of the fields on *Nutstat*'s main screen. A field can be added or removed from the main screen simply by checking or unchecking the check box next to the field name. The reference standard that is used to calculate the statistics can also be changed on this tab. If the reference standard is changed, the user will be prompted if they want to update all of the existing records with statistics calculated with the newly selected reference standard. WHO/CDC and NCHS statistics are stored in separate columns in the data table so that statistics calculated with one reference standard will not replace statistics calculated with the other reference standard. (Note: the NCHS reference set expected in the year 2000 is not yet available.) The tab labeled UNITS allows the user to switch between Metric and English units for the data entry fields. This tab also allows the user to select the format for the date fields and whether the age should be entered in months or years and months.

The tab labeled GRAPH allows the user to select which graph will be produced when the GRAPH button is clicked on the main screen. If a graph is dependent on a data from a field that is not being displayed on the main screen, the graph will not be included in the list.

PERCENTILE LINE RESOLUTION controls the number of points that are used to create the growth curves. The higher the resolution, the smoother the curves, but the longer it takes to produce the graphs. On slower computers, you may want to select a lower percentage of points to display. If the SHOW BEFORE GRAPHING check box is checked, the graph options will be displayed every time the GRAPH button is clicked on the main screen.

Printing From Nutstat

Nutstat uses templates to determine the format of the output when printing. The FILE:PRINT menu item will use the last selected template to produce the printout. Printed output can be previewed by selecting FILE:PRINT PREVIEW. The Print Preview screen will display the data associated with the current ID Number. The Print Setup screen is accessed by way of the CUSTOMIZE button on the Print Preview screen. The Print Setup screen consists of three sections separated by tabs. The GENERAL tab enables the customization of the information contained in the beginning and end of the document. Text entered in the HEADER text area will be displayed in the top right corner of the first page of the document. The PATIENT INFORMATION list box will display all of the checked items directly below the Header information. An image can be added by entering the path name of a Bitmap, Icon, or Windows Metafile image in the IMAGE FILE text box. The image will be displayed in the upper right corner of the first page of the document. The size of the image can be adjusted by changing the value in the IMAGE SIZE combo box. A Text or Rich Text file with additional information can be appended to the document by entering the path name of the file ADDITIONAL INFORMATION FILE

The GRAPHS tab allows the addition of graphs to the document. Checking the check box next to a graph will cause it to be inserted into the document. A title or description can be added to a graph by selecting it in the list and adding the title and/or description in the GRAPH TITLE and GRAPH DESCRIPTION text boxes respectively.

The TABLE tab allows the addition of a table. Fields are added to the table by moving the field name from the AVAILABLE FIELDS list to the SELECTED FIELDS list. The text entered in the TABLE TITLE field will be added as the title for the table.

The SAVE and SAVE AS buttons on the bottom of the screen are used to save the current settings on the screen in a template file. The OPEN button is used to open a previously saved template. Clicking the OK button will save the current settings on the screen to the default template, refresh the Print Preview screen with the new settings and close the Print Setup screen. Since the OK button always saves the settings to the default template, it is only necessary to use the Save and Save As buttons if multiple templates are desired. Clicking the Cancel button in the Print Setup screen will close the screen without refreshing the Print Preview screen or saving the changes to the default template.

Generating Reports With Nutstat

The *Nutstat* Report Generator is reached from the FILE:REPORT GENERATOR menu item. The first screen to display after selecting the menu item will be the REPORT

GENERATOR – SOURCE screen. This screen has two textboxes that capture the template path name and the data source. The template path name can be entered directly into the text box or located by clicking the BROWSE button.

The Data Source text box is read only. The data source can be changed by clicking the CHANGE button. This will display the DATA SOURCE screen, which is similar to the FIND screen. The data source can be selected based on ID Numbers, Birth Dates, Measurement Dates or a combination thereof. The TEST button will display the records in the data source in a grid. Clicking OK will refresh the Data Source text box on the REPORT GENERATOR - SOURCE screen with the new data source and close the DATA SOURCE screen. Clicking the CLOSE button on the menu bar will close the Data Source screen without updating the data source.

Clicking the OK button on the REPORT GENERATOR –SOURCE screen will display the DATA REPORTER screen. This screen allows customizing the report. The LAYOUT WINDOW toolbar button pops-up a window that lists all of the fields in the data source. The fields can be added and grouped in the report simply by dragging and dropping them into the appropriate list. The PROPERTIES toolbar button pops up a window that allows several formatting properties to be set. The report template can be saved by way of the SAVE or SAVE AS item in the File menu.

External Data Features of Nutstat

Nutstat has two features that deal with external data--ADD STATISTICS and IMPORT. The purpose of the Add Statistics feature is to add statistical data to an existing table. The purpose of the Import feature is to import data from an existing table into a new table that has the data structure that is required for Nutstat to use it. The table can be an Epi 2000 table or a table from Microsoft Access.

ADD STATISTICS and IMPORT are accessed via the FILE:EXTERNAL DATA menu item. The first two screens to be displayed after selecting either feature will capture the database and table name for the table to be imported or added to. The third screen that is displayed will capture all of the information needed to process the incoming table.

The frame with the label SELECT THE FIELDS TO USE is common to both ADD STATISTICS and IMPORT. It is used to link the fields in the table with the type of data they contain. The list box labeled INPUT FIELD contains all the fields in the source table. The list box labeled FIELD TYPE contains all the field types that the process can use. To link a field to a field type, select the field and field type in the list boxes and click the link button. For some field types, a UNITS dialog box may pop up to capture the unit of measurement that the data in the field is in. After the unit is selected (if necessary) the field, field type and unit of measurement will be added to the Linked Fields grid.

The only field types that are required are the SEX and DATE OF MEASUREMENT fields (marked by asterisks) when importing. None of the other field types are required, but, if a statistic is dependent on a field that is not linked, it cannot be calculated.

The frame labeled SELECT THE REFERENCE STANDARD TO USE is common to both IMPORT and ADD STATISTICS. It is used to select the reference standard to use when processing the data in the table. Currently, only the WHO/CDC 1977/1985 reference is enabled.

The frame labeled SELECT THE STATISTICS TO ADD is unique to the ADD STATISTICS feature. It contains check boxes that indicate whether a statistic should be added to the table. The check boxes are only enabled if the required field types for the statistics have been linked. Hovering the mouse pointer above a check box will display a tool tip that lists the field types required for the respective statistic.

The OUTPUT DATABASE and OUTPUT TABLE text boxes are unique to the Import feature. They capture the path and name of the database and the name of the table that the data will be imported into. If the database or table does not exist, it will be created when the Import is preformed.

The PROCESS' button is common to both IMPORT and ADD STATISTICS. Clicking this button starts the processing of the data. A dialog box with a progress bar will display after clicking the process button to indicate the progress of the process.

Nutstat as Part of an Epi Info 2000 Questionnaire View

NutStat can be run from within an Epi Info questionnaire view as though it is a related table. *Nutstat* is represented by a button on the questionnaire that can be clicked to bring up the nutritional entry screen.

To set up the relationship and the button, right click on the screen of a questionnaire in *MakeView*, enter a name for the button to be created (“Growth,” for example), and then click on the button labeled CREATE RELATED VIEW. In the next dialog, selection criteria can be entered so that the button is active only under certain conditions, or it can be made active at all times. A reasonable condition might be that an AGE field contain a value between 0 and 18 years, since most of the calculations in *NutStat* are limited to this age group.

Overview of 1985 CDC/WHO Growth Reference Curves

The following sections were written by Kevin Sullivan, PhD, Department of Pediatrics, School of Medicine, and Department of Epidemiology, Rollins School of Public Health, Emory University, Atlanta, Georgia and Nathan Gorstein, World Health Organization, for the Epi Info 6 manual.

The anthropometric calculations described in this chapter are based on the growth reference curves developed by the National Center for Health Statistics (NCHS) and CDC using data from the Fels Research Institute and US Health Examination Surveys.¹ These growth curves are recommended by the World Health Organization (WHO) for international use.²

NCHS is in the process of developing new growth reference curves for the US. When these become available, they will be incorporated into the *NutStat* program so that both sets of curves will be available.

To calculate anthropometric indices, information is needed on each individual's sex, age, weight, and height. From these data it is possible to form different indices, including those that relate to height-for-age (HA), weight-for-age (WA), and weight-for-height (WH). These indices can be expressed in terms of Z-scores, percentiles, and percent of median relative to the international growth reference population mentioned above. The following abbreviations will be used throughout this chapter:

HAP	Height-for-Age Percentile
HAZ	Height-for-Age Z-score
HAM	Height-for-Age percent of Median
WAP	Weight-for-Age Percentile
WAZ	Weight-for-Age Z-score
WAM	Weight-for-Age percent of Median
WHP	Weight-for-Height Percentile
WHZ	Weight-for-Height Z-score
WHM	Weight-for-Height percent of Median

Interpretation and Uses of Anthropometry

Anthropometry can be used to assess nutritional status at both the individual and the population level. Ideally, individuals should have several weight and height measurements over time so that growth velocity can be assessed. A decline in an individual's anthropometric index from one point in time to another could be an indication of illness and/or nutritional deficiency that may result in serious health outcomes. In some situations, a single set of measurements may be used for screening populations or individuals to identify abnormal nutritional status and priority for

treatment.

At the population level, data are most commonly available from cross-sectional surveys in which the prevalence of low anthropometric indices can be assessed by determining the proportion of the population that falls below a cutoff value. In addition, the mean or median anthropometric value of a population can be compared with the reference value to assess the status of the study population relative to the reference population.

The two preferred anthropometric indices for determining nutritional status are WH and HA, as these discriminate between different physiological and biological processes.^{2,3} Low WH is considered an indicator of wasting (i.e., "thinness") and is generally associated with failure to gain weight or a loss of weight. Low HA is considered an indicator of stunting (i.e., "shortness"), which is frequently associated with poor overall economic conditions and/or repeated exposure to adverse conditions. The third index, WA, is primarily a composite of WH and HA, and fails to distinguish tall, thin children from short, well-proportioned children.

The distribution of the indices can be expressed in terms of Z-scores, percentiles, and percent of median. Z-scores, also referred to as standard deviation (SD) units, are frequently used. The Z-score in the reference population has a normal distribution with a mean of zero and standard deviation of 1. For example, if a study population has a mean WHZ of 0, this would mean that it has the same median WH as the reference population. The Z-score cutoff point recommended by WHO, CDC, and others to classify low anthropometric levels is 2 SD units below the reference median for the three indices. The proportion of the population that falls below a Z-score of -2 is generally compared with the reference population in which 2.3% fall below this cutoff. The cutoff for *very* low anthropometric levels is usually more than 3 SD units below the median.

Percentiles, or "centiles," range from zero to 100, with the 50th percentile representing the median of the reference population. Cutoff points for low anthropometric results are generally < 5th percentile or < 3rd percentile. In the reference population, 5% of the population falls below the 5th percentile; this can be compared with the proportion that falls below this cutoff point in the study population.

The calculation of the percent of median does not take into account the distribution of the reference population around the median. Therefore, interpretation of the percent of median is not consistent across age and height levels nor across the different anthropometric indices.²

Traditionally, in the United States and some other countries, percentiles are used as cutoff points. In other parts of the world, either Z-scores or percent of median are used, although WHO favors the use of Z-scores.²

Z-scores and percentiles are directly related. Both rely on the fitted distributions of the indices across age and height values and are consistent in their interpretation across

anthropometric indices. Z-scores are useful because they have the statistical property of being normally distributed, thus allowing a meaningful average and standard deviation for a population to be calculated. In addition, Z-scores have a greater capacity to determine the proportion of a population that falls below extreme anthropometric values than do percentiles.

Percentiles are useful because they are easy to interpret (e.g., in the reference population 3% of the population falls below the 3rd percentile). Percentiles, however, are generally not normally distributed in either the reference or the study populations.

The more common cutoff value used is < -2 SD. The prevalence of < -2 SD can be compared with other countries as shown in Table 1.³ For example, in a survey of children, if the prevalence of weight-for-height < -2 SD is found to be 16.7%, this would be considered to be a *very high* prevalence of low WH. If the prevalence of low height-for-age is found to be 18.9%, this would be a low prevalence of stunting. The prevalence of low anthropometric indices should be presented by one-year intervals for children less than six years of age, or, if age is unknown, for children < 85 centimeters compared with those ≥ 85 centimeters, which approximates comparing the children < 2 years of age to those ≥ 2 years.

Table 1: Prevalence of low anthropometric values (< -2 SD) compared to other surveys for children five years of age or less

Relative Prevalence of Low Anthropometric Values				
Index	Low	Medium	High	Very High
Low WH	$< 5.0\%$	5.0-9.9%	10.0-14.9%	$\geq 15\%$
Low HA	$< 20.0\%$	20.0-29.9%	30.0-39.9%	$\geq 40.0\%$
Low WA	$< 10.0\%$	10.0-19.9%	20.0-29.9%	$\geq 30.0\%$

Limitations of Growth Reference Curves

HA and WA indices can be calculated for individuals from birth up to 18 years of age. WH indices are calculated for males to 138 months (11.5 years) of age and less than 145 cm (57 inches) and for females to 120 months (10 years) of age and less than 137 cm (53 inches). WH cannot be calculated for children less than 49 cm (19.3 inches). For children less than 2 years of age, recumbent (i.e., lying down) length measurements are assumed; for children 2 years of age and older, height refers to standing height.

No anthropometric indices are calculated if sex is unknown or miscoded because there are separate growth reference curves for males and females. If weight is unknown, only HA will be calculated; if height is unknown, only WA will be calculated; and if age is unknown, only WH will be calculated. When age is unknown, children shorter than 85 centimeters are assumed to be less than 2 years of age; otherwise, WH is calculated with the assumption that the child is 2 years of age or older.

How to Reduce Anthropometric Errors

Below are some basic steps to follow to ensure that age, weight, and height data are collected accurately.

- Make sure the equipment is correctly calibrated on a regular basis.
- Thoroughly train those who collect the data.
- To reduce errors on the child's age, collect information on both the child's age and the dates of birth and measurement. The year of birth is frequently given incorrectly. Compare the calculated age with the age provided by the child's caretaker. If there is a large discrepancy between the two age values, the age provided by the caretaker is probably closer to the true value. Check the year of birth and see how the anthropometric indices change if you correct the year of birth to correspond with the stated age.
- After the age, sex, weight, and height information are collected, check the data against a growth chart or by calculating the anthropometric indices on a computer. Children with extreme values should be remeasured.
- For research projects, data should be entered twice and compared or otherwise confirmed as correct.

What the Anthropometric Calculation Program Does

SEX - coded as 1/M/m for males, 2/F/f for females

AGE - in months

WEIGHT - in kilograms

HEIGHT - in centimeters

The program performs the calculations and returns the following values:

	Percentiles	Z-scores	Percent of Median
Height-for-Age	HAP	HAZ	HAM
Weight-for-Height	WHP	WHZ	WHM
Weight-for-Age	WAP	WAZ	WAM

A record FLAG, coded 0 to 7, described below.

The first nine fields contain the results of the anthropometric calculations. For the Z-

scores, a code of 9.99 means that the index could not be calculated because of missing data or data values that were out of the appropriate range. An example of the latter would be an age of 18 years or older. A code of 9.98 for Z-scores denotes that the Z-score was greater than or equal to 9.98 and most likely indicates an error in measurement. For percentiles and percent of median, a similar coding scheme is used (99.9 and 99.8 for percentiles and 999.9 and 999.8 for percent of median, respectively).

A tenth field, the record FLAG field, is used to identify records where there are missing data points or a strong likelihood that some of the data items are incorrect (based on extreme Z-scores). The criteria for "flagging" an anthropometric index are as follows:

Index	Minimum	Maximum
HAZ	-6.00	+6.00
WHZ	-4.00	+6.00
WAZ	-6.00	+6.00

Two additional criteria for "flagging" a record are combinations of data items:

(HAZ > 3.09 and WHZ < -3.09) *or* (HAZ < -3.09 and WHZ > 3.09)

It is recommended that all "flagged" records be verified for accuracy. Common errors include incorrect data entry, incorrect age/dates, weight or height measurements entered incorrectly or in the wrong units, and missing/blank data. When anthropometric data are being analyzed in the Epi Info *Analysis* program or elsewhere, it is recommended that certain indices be set to missing (and therefore excluded from analyses) based on the coding in the FLAG field (Table 2). Note that when a Z-score is flagged, the corresponding percentiles and percent-of-median values are also flagged.

Table 2: Record flag coding scheme

Flag Code	Index Flagged			Notes
	HAZ	WHZ	WAZ	
0				No indices flagged
1	Y			Only HAZ flagged
2		Y		Only WHZ flagged
3	Y	Y		Both HAZ and WHZ flagged
4			Y	Only WAZ flagged
5	Y		Y	Both HAZ and WAZ flagged
6		Y	Y	Both WHZ and WAZ flagged
7	Y	Y	Y	All three indices flagged

Y=Index flagged, blank means index not flagged.

Interpretation of the flags is as follows:

Flag 0: This means that none of the indices were flagged. However, this does not necessarily mean the information is correct. Either sex, age, weight, or height could be

incorrect but not extreme enough to be flagged.

Flag 1: HA is flagged but not WH or WA. This could be an extremely short or tall individual. Assure that the height information entered onto the computer file is correct. If height is incorrect, then WHZ would generally be close to -3.09 or 3.09 (a WHZ value beyond these would produce a flag error number 5). The other alternative is that the age information is incorrect, which would make the WAZ extreme (near -6 or 6).

Flag 2: WH is flagged but HA and WA are not. First, check the age and height of the child and make sure they are within the limits described in the section *Limitations of Growth Reference Curves*. If the child is within the age and height limitations, then either height or weight may be incorrect. If height is incorrect, then HAZ would be expected to be near an extreme value (but not extreme enough to be flagged), and if weight is incorrect, then WAZ would be close to an extreme value (but not extreme enough to be flagged). Finally, this could truly be an extremely thin or obese child.

Flag 3: HA and WH are both flagged but WA is not. This is an indicator that height may be incorrect or missing.

Flag 4: WA is flagged but not HA or WH. If the weight is incorrect, then WHZ would be near an extreme value (but not extreme enough to be flagged), and if age is incorrect, then HAZ is likely to be near an extreme value (but not extreme enough to be flagged).

Flag 5: HA and WA are flagged but not WH. This is an indication that the age information is incorrect, missing, or out of range.

Flag 6: WH and WA are flagged but not HA. This is an indication that weight is likely to be incorrect or missing.

Flag 7: All three indices are flagged. This can occur if sex is unknown or incorrectly coded; or at least two of the following are missing, incorrectly coded, or beyond the limitation of the growth curve: age, weight, or height.

Other Considerations

As mentioned above, the subroutine that calculates anthropometry assumes that age is in months, sex is coded as 1/M/m for males and 2/F/f for females, weight is in kilograms, and height is in centimeters. However, with Epi Info, you can calculate age from two dates (i.e., date of birth and date of measurements) and convert US measurements (e.g., inches and pounds) entered on the screen to metric within a .CHK file (described in more detail later in this chapter).

When possible, it is always preferable to have age calculated from the birth date and the

date of the measurement(s). The reference curves are based on "biologic" age rather than calendar age. Biologic age in months divides the year into 12 equal segments as opposed to calendar age in which months have from 28 to 31 days. Although this makes little difference in older children, it can have an effect on the anthropometric calculations for infants. To calculate biologic age, the number of days between the two dates is calculated by the program. The age in days is divided by 365.25 and then multiplied by 12. Entering age by rounding to the nearest month and/or the most recently attained month can have a substantial effect on the anthropometric calculations, especially for infants.⁴

Other Anthropometric Software and Sources of Information

For additional information on the use and interpretation of anthropometry, please refer to the articles by: WHO Working Group on the Purpose, Use, and Interpretation of Anthropometric Indicators of Nutritional Status;² Gorstein et al.,³ Dibley et al.;⁵ Waterlow et al.,⁶ and Beaton et al.⁷ To obtain additional information on anthropometric software, please contact one of the following:

Division of Nutrition
Center for Chronic Disease Prevention and Health Promotion
Centers for Disease Control and Prevention
1600 Clifton Road NE, MS A08
Atlanta, GA 30333 U.S.A.
Nutrition Unit
World Health Organization
1211 Geneva 27
Switzerland

Acknowledgments

Special thanks to Phillip Nieburg, M.D., M.P.H., Norman Staehling, M.S., Ronald Fichtner, Ph.D., and Fredrick Trowbridge, M.D., Centers for Disease Control and Prevention, for their assistance.

References

1. Dibley MJ, Goldsby JB, Staehling NW, Trowbridge FL. Development of normalized curves for the international growth reference: historical and technical considerations. *Am J Clin Nutr* 1987;46:736-48.
2. WHO working group. Use and interpretation of anthropometric indicators of nutritional status. *Bull WHO* 1986;64:929-41.
3. Gorstein J, Sullivan K, Yip R, de Onis M, Trowbridge F, Fajans P, Clugston G. Issues in the assessment of nutrition status using anthropometry. *Bull WHO* 1994 (in press).
4. Gorstein J. Assessment of nutritional status: effects of different methods to determine age on the classification of undernutrition. *Bull WHO* 1989;67:143-50.
5. Dibley MJ, Staehling N, Nieburg P, Trowbridge FL. Interpretation of Z-score anthropometric indicators derived from the international growth reference. *Am J Clin Nutr* 1987;46:749-62.
6. Waterlow JC, Buzina R, Keller W, Land JM, Nichaman MZ, Tanner JM. The presentation and use of height and weight data for comparing the nutritional status of groups of children under the age of 10 years. *Bull WHO* 1977;55:489-98.
7. Beaton G, Kelly A, Kevany J, Martorell R, Mason J. Appropriate uses of anthropometric indices in children: a report based on an ACC/SCN workshop. ACC/SCN Nutrition Policy Discussion Paper no. 7, 1990.
8. Mei Z, Grummer-Strawn LM, de Onis M, Yip R. The development of a MUAC-for-height reference, including a comparison to other nutritional status screening indicators. *Bull WHO* 1997;75:333-342.

Commands



Detailed Descriptions of Commands in the Epi Info Programming Language

Overview

Epi Info 2000 is easy to operate in interactive mode, but more complex operations require saving the steps as programs. Programs (similar to “scripts” in some software) can be used to set up Menus, guide and limit the data entry process, restructure data, and do analysis. Template files save the settings for Graphs and Maps so that similar graphics can be produced more than once.

In *MakeView* and in *Analysis*, the process of programming consists of interacting with a series of dialogs that produce the actual program statements. Experienced users may want to edit the statements or type them directly in the program editor. For this reason, the details of command syntax are provided in this chapter. A definition of each command and its operation is given. Because a single command such as EXECUTE may be found in *MakeView*, *Analysis*, and the Menu environments, each command is presented in a single section of the text in this chapter, with notation of differences that may exist in its implementation in the various programs. Some commands are only available in one or two of the programs. At the top of the description of each command, the programs in which it is available are clearly marked—as *Menu*, *Check*, or *Analysis*. *Check* commands are saved in *MakeView* and executed in *Enter*. *Menu* commands are inserted in .MNU files with a word processor and executed by the *Epi2000* Menu. *Analysis* commands are generated, edited, and executed in *Analysis*.

Functions and operators appear within commands and are used for such common tasks as extracting a year from a date, combining two numeric values, or testing logical conditions. Because there are many functions and operators, they are presented separately in the Functions and Operators Chapter.

Epi Info 2000 Commands

<p><u>PROGRAMS</u> ANALYSIS.EXE ENTER.EXE EPI2000.EXE EPIMAP.EXE MAKEVIEW.EXE NUTSTAT.EXE</p> <p><u>VARIABLES</u> ASSIGN DEFINE UNDEFINE</p> <p><u>DATA</u> CANCEL SELECT CANCEL SORT COPY IF THEN ELSE IMPORT FROM EPI6 LIST MERGE READ RECODE RELATE SELECT SORT WRITE</p> <p><u>USER INTERACTION</u> DIALOG HELP MENU</p>	<p><u>OUTPUT</u> FREQ GRAPH MAP MATCH MEANS TABLES HEADER TYPE</p> <p><u>STATISTICS</u> KMSURVIVAL LOGISTIC REGRESS</p> <p><u>MOVEMENT</u> AUTOSEARCH GOTO</p> <p><u>SYSTEM</u> EDITVIEW EXECUTE PRINT PRINT VIEW</p> <p><u>SCREEN</u> CLEAR DIALOG HELP HIDE UNHIDE</p> <p><u>PROPERTIES OF VIEW</u> CODES COMMENT LEGAL ENCRYPTION LEGAL MUSTENTER PASSWORD RANGE READ ONLY REPEAT SOUNDEX</p>	<p><u>MENU</u> BUTTON MENUITEM POPUP SCREENTEXT</p> <p><u>COMMAND BLOCKS</u> @@ BLOCK COMMENTS EXECUTE EXIT HELP MENU PICTURE</p> <p><u>SYSTEM CONFIGURATION</u> AUTOSEARCH DOSWIN QUIT (EXIT) RUNPGM SET SET BUTTONS SET PICTURE SETWORKDIR SYSINFO</p>
---	--	--

Epi Info 2000 Programs: General Features

EPI2000.EXE

Program

Description	The Epi Info Menu
Command Line Parameters	EPI2000.exe Epi2000 <Mnu File> <Image>

	Syntax Element	Description
Comments	<Mnu File>	Path and Name of a menu (.MNU) file
	<Image>	Path and name for a valid Image

Buttons, menu items, screen text, background pictures, and procedures to be executed from the menu are controlled by the MNU file and can be programmed by the user.

The image can be in GIF, JPG, or BMP format. They can be located in different directories.

Examples	Epi2000.MNU and Surveil.MNU, supplied with the programs.
----------	--

MAKEVIEW.EXE

Program

Description	The Design Environment for View Questionnaires
Command Line Parameters	Makeview <Project>:<View>

	Syntax Element	Description
Comments	<Project>	A valid path and MDB database
	<View>	A valid view in the project specified

This is an ActiveX executable module that can be run from within the **Enter** program and vice versa. Views are saved as tables with a specific format within an MDB (Microsoft Access) database.

Examples	Makeview "C:\epi2000\Sample>MDB":Oswego
----------	---

ENTER.EXE

Program

Description	The Data Entry Program						
Command Line Parameters	ENTER <Project>:<View>						
Comments	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Syntax Element</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td><Project></td> <td>A valid path and MDB database</td> </tr> <tr> <td><View></td> <td>A valid view in the project specified</td> </tr> </tbody> </table>	Syntax Element	Description	<Project>	A valid path and MDB database	<View>	A valid view in the project specified
Syntax Element	Description						
<Project>	A valid path and MDB database						
<View>	A valid view in the project specified						
Examples	ENTER C:\epi2000\Sample.MDB:Surveillance						

ANALYSIS.EXE

Program

Description	A general-purpose data management and <i>Analysis</i> program that reads many types of files and will produce lists, frequencies, tables, epidemiologic statistics, graphs, and maps.								
Command Line Parameters	ANALYSIS '<Filename>'.<PGM>'								
	or								
	ANALYSIS PGMNAME=<PgmName> [VIEWNAME=<ViewName>]								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Syntax Element</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td><Filename></td> <td>MDB path and name (in single quotes)</td> </tr> <tr> <td><PGM></td> <td>Program to be executed (in double quotes)</td> </tr> <tr> <td><View></td> <td>View name</td> </tr> </tbody> </table>	Syntax Element	Description	<Filename>	MDB path and name (in single quotes)	<PGM>	Program to be executed (in double quotes)	<View>	View name
Syntax Element	Description								
<Filename>	MDB path and name (in single quotes)								
<PGM>	Program to be executed (in double quotes)								
<View>	View name								

Comments

Analysis can be run from the menu or from other programs. A program to be run and an optional Viewname may be specified on the command line. If a program is given, it can be one that is stored in a database or a text file with the PGM extension. For example:

```
ANALYSIS 'SURVEILLANCE.MDB':"WEEKLY"
Or ANALYSIS PGMNAME=SURVEILLANCE.MDB:"Weekly"
```

might run a weekly report program stored in the database called SURVEILLANCE. VIEWNAME= <viewname> can be added.

```
ANALYSIS 'C:\Epi2000\LASTPGM.PGM'
```

runs the program automatically saved from the program editor on last exit from *Analysis*.

From the Menu

Values can be passed from the menu directly to *Analysis*. The syntax for executing a program is:

```
EXECUTE Analysis '<Filename>': "<Program>"
```

Examples

```
Analysis 'SAMPLE.MDB':"STATISTICS"
Analysis PGMNAME='C:\Epi2000\Sample.MDB':"Statistics"
(To READ a view before loading the program, we could add:
VIEWNAME='C:\epi2000\Sample.MDB':viewOSWEGO)
From the Menu
EXECUTE ANALYSIS 'SAMPLE.MDB':"STATISTICS"
```

NUTSTAT.EXE

Program

Description	A program for Nutritional Anthropometry	
Command Line Parameters	<UniqueID of parent record>	
	Syntax Element	Description
Comments	Can be run from a View in <i>Enter</i> or from other databases by passing the record ID of a parent record. Records in <i>NutStat</i> are for body measurements on a particular date.	
Examples		

EPIMAP.EXE

Program

Description	A mapping program built around MapObjects from Environmental Systems Research Institute, Inc. (ESRI)	
Command Line Parameters	EPIMAP <Map File >	
	Syntax Element	Description
	<Map File>	Path and Name of .MAP file
Comments	When run as an ActiveX executable object, many properties can be set. NOTE: CDC's license for MapObjects does not permit <i>EpiMap</i> to be used independently from Epi Info.	
Examples	Epimap Mexmap95.Map	

Special Features of Check Commands

Check commands are optional, and not all fields need to have blocks of commands in a particular view.

Check commands must be placed in a named block of commands beginning with the name of a variable in the database and ending with the word END. Special blocks are provided to execute commands before or after entering a View, Record, or Page. These are:

BEFORE VIEW, AFTER VIEW, BEFORE RECORD, AFTER RECORD, BEFORE PAGE, and AFTER PAGE.

Comments, preceded by an asterisk ("*"), may be placed within blocks of commands. For commands not having a "Command...end" structure, the line continuation character ("\" at the end of the line) is used to indicate that a command continues on the next line.

Commands in a block are activated either before or after an entry is made in the field. The default is that the commands are performed after an entry has been completed with <Enter>, <PgUp>, <PgDn>, or <Tab>, or another command causes the cursor to leave the field.

Check commands for each field are stored in the VIEW in a record associated with a particular field or with one of the special BEFORE/AFTER blocks mentioned above.

The commands are inserted automatically through interaction with the PROGRAM dialog boxes, chosen from a tabbed display of commands. Text versions of the

commands appear in the Program Editor as they are generated by the dialogs, and can be edited there if desired.

Menu Check Analysis

@@

Command

X

Description

Replaces a variable name with its current value

Syntax

@@<VarName>

Syntax Element	Description
<VarName>	Variable name

Comments

The name of a variable created with ASSIGN or DIALOG can be embedded in a menu command, a MenuItem, Screentext, or a Button description. When the command is executed, the Value of the variable will be substituted for @@VarName.

Users of Epi Info 6 will note that that @@ replaces @ to avoid confusion with Internet addresses that frequently contain a single @ sign.

Examples

```
Assign FavoriteURL = "WWW.CDC.GOV"
LINKTO @@FavoriteURL
```

If all goes well (and you have an Internet connection and browser), these two commands will connect to the CDC web page. Of course, LINKTO WWW.CDC.GOV will do the same thing, but for repeated use, allowing users to set the variable value, or passing a value to another program, the embedded variable approach can be very helpful.

Program-Specific
Features

Menu

Available for use in menu command blocks

This is one way to assign a value to a variable. The Value is placed in the file called EPIINFO.INI; therefore all variables created in the menu file will be permanent in any of the Epi Info programs. In the menu environment, there is no need to first DEFINE a variable; they are defined automatically by the ASSIGN or DIALOG commands, and will accept either text or numeric data. Text should be enclosed in quotation marks, although this is not always necessary. Variable names cannot contain spaces or punctuation, but both are permitted in values.

Functions described in the Functions Chapter for *Analysis* and Check commands are not available in the Menu.

Enter

MakeView can define variables to be used by *Enter*. Those variables will be places in a specific area of the check code called Defined Variables. Variables can have any of the three scopes (standard, global, and permanent) available in Epi Info 2000. Global variables will retain values while moving from one table to a related one. Permanent variables are stored (in the EPIINFO.INI file) even after *Enter* closes.

Analysis

If the right side of the assignment does not contain a data variable (one in a database table), or a variable that depends on a data variable, then the assignment will be made immediately. Thus:

```
DEFINE YEAR PERMANENT
YEAR = 2000
```

However:

```
DEFINE INCUBATION
INCUBATION = ONSETDATE-EXPOSUREDATE
```

contains two VIEW variables, ONSETDATE and EXPOSUREDATE. INCUBATION will only be calculated during processing of the current dataset. It will be calculated for each record during processing and may be used like a dataset variable in TABLES, FREQ, and GRAPH commands.

Prior to and after processing a dataset, INCUBATION will have a “missing” value, although it could be assigned a value with another statement such as INCUBATION = 999.

The value is calculated each time a record that meets the conditions of SELECT is read from the dataset. Any legal expression can be used that combines variables or specific values and operators like +, -, *, /, =, <, >, AND, OR, and NOT. See OPERATORS for the complete list. Standard variables that depend on database fields must be saved to a table using WRITE before they can be edited using LIST UPDATE.

Menu Check Analysis

AUTOSEARCH

Command

X

Description

AUTOSEARCH causes *Enter* to search for one or more matching records. If a match is found, the user can choose to display and edit the matching record(s) or to ignore the match and continue to enter the current record.

Syntax

AUTOSEARCH <VarName(s)>

Syntax Element	Description
<VarName(s)>	A group of variables to look for.

Comments

If more than one record is found, the results are displayed in a spreadsheet format that allows several records to be displayed. To see more than one screen of such matches (all the SEX = "M" records, for example) press <PgDn> to see the screens after the first. When matches have been displayed, a single row can be selected by double clicking to the left of the first field or by using the up and down arrow keys, and a particular record selected for editing by pressing <Enter> or selecting OK. To avoid editing any of the matches, press <Esc> or the Cancel button. The fields displayed as the result of a LIST search are determined as follows:

1. If a single field is the key, that field and as many other fields as possible are displayed.

2. If there are multiple keys, these are displayed first and then as many of the remaining fields as possible are shown, starting with the first. AUTOSEARCH will use the Soundex keys on all search fields that have them.

Examples

AUTOSEARCH can be given a list of fields to use for the search, rather than relying on the order of the fields in the view. If a view has the fields:

```
FIELD1 ...
FIELD2 ...
FIELD3 ...
FIELD4 ...
FIELD5
```

```
FIELD3
  AUTOSEARCH FIELD2 FIELD3
END
```

then AUTOSEARCH will ignore the value of FIELD1 and search only on the values of FIELD2 and FIELD3.

		Menu	Check	Analysis
BEEP				
Command			X	X
Description	Generates a sound.			
Syntax	BEEP			
Comments				
Examples	<pre>IF not exists("c:\epi2000\Sample.MDB")= (+) then BEEP Dialog "Please check that the file exists and you have permission to access it" TITLETEXT="File not found" END</pre>			

Menu Check Analysis

BUTTON(S)

Command

X

Description

Creates a button on the screen with the caption ItemText and the locations given as percentages of screen width and height. When the button is clicked with the mouse (left mouse click), the commands in the CommandBlock named are executed. If a text tip is given, this appears when the mouse cursor hovers over the button for several seconds.

Syntax

BUTTON "<Caption>", <Block>, <Xlocation>,<YLocation>,"<Dynamic text>"

Syntax Element	Description
<Caption>	Text to be displayed in the button
<Block>	The name for the block to be executed
<X-Location>	A number from 1 to 99 representing the distance from the left side of the window in percentage points.
<Y-Location>	A number from 1 to 99 representing the distance from the top of the window in percentage points.
<Dynamic Text>	Text to be displayed when the cursor is pointing to that button

Comments

Hot keys may be specified by preceding them with an ampersand (&), in which case the Alt key plus the hot key will activate the button. Hot keys should not be duplicated between POPUP items (menu headings) and BUTTONS.

Examples

```
*BUTTONS
*Begin
BUTTON "Ma&keView", MakeView, 5,50,"Make or Edit a
Questionnaire"
BUTTON "E&nter Data", Enter, 5, 60, "Enter and store data,
using\ a View"
```

Menu Check Analysis

CANCEL SORT/SELECT

Command X

Description Cancels a previous SORT or SELECT command.

Syntax SORT
 SELECT

Comments Cancel sort and Cancel select will automatically close the current output file.

Examples Read 'C:\EPI2000\SAMPLE.MDB': viewOswego
 Select ill = (+)
 List *
 Select
 List *

Menu Check Analysis

CLEAR

Command X

Description CLEAR sets the field named to the missing value, as though the field had been left blank. It is useful to clear a previous entry after an error has been detected, for example. More than one field may be specified. CLEAR will frequently be followed by a GOTO, to put the cursor in position for further entry after an error.

Syntax **CLEAR** <VarName(s)>

Syntax Element	Description
<VarName(s)>	A valid name for a variable

Comments CLEAR will delete the value only for the current record. Clear cannot be used in GRID Tables.

Examples If EndDate < StartDate then
 CLEAR EndDate
 GOTO EndDate
 END IF

Program-Specific Features **Analysis**

To clear a variable in *Analysis*, it is necessary to [ASSIGN](#) null (.) to it

	Menu	Check	Analysis
CLOSEOUT			
Command			X
Description	Closes the current output file.		
Syntax	CLOSEOUT		
Comments	Closeout is the standard way to end an HTM file, In some cases when the output is too long, the browser will require a significant amount of time to display the output. To avoid this problem try to avoid large output files.		
Examples	ROUTE Outbreak1.HTM Tables * ILL Tables AGE SEX ILL MEANS AGE ILL CLOSEOUT		
Program-Specific Features	This command should be used before moving to a different output file, whether via the Results Library or the browser's BACK button.		

	Menu	Check	Analysis
CMD			
Command			X
Description	The CMD command can be used to define a block of commands that will act as a unit, similar to a subroutine or procedure in other languages.		
Syntax	CMD <COMMAND NAME> [Commands] [Commands] END		

Syntax Element	Description
<CommandName> [Commands]	Execute the defined block of instructions At least one valid Epi2000 command

Comments Whenever <CommandName> appears elsewhere in the program, the block of commands is executed.

Examples
 CMD HELLO
 DIALOG "Welcome to Epi2000" TITLETEXT = "Greetings"
 END
 HELLO
 HELLO
 HELLO

Program-Specific Features Use of the command generator within a CMD is limited, as these commands are not actually run until CommandName is encountered in a program. However, any command that will be legitimate at the time it is actually run can be put into a CMD block using the text editor.

Menu Check Analysis

COMMAND BLOCKS

Command **X**

Description	Command blocks called from the MENUITEM and BUTTON commands	
Syntax	<Block Name> Begin Command Command End	
	Syntax Element	Description
	<Block Name>	A Name containing only text characters and/or digits. Special characters such as (, #, \$, &, and blank spaces are not allowed as part of the block name).
	Begin/ End	Delimiters for the command block.
Comments	Commands in Command Blocks are executed when the user chooses a menuitem or button that refers to the block. If the begin and end statements are missing, results are unpredictable.	
Examples	MAKEVIEW Begin EXECUTE MakeView.EXE End Menu	
Program-Specific Features	The main menu will create a batch file to execute DOS commands; therefore, they will be executed at a single time. Batch files containing both windows and DOS command are not allowed.	

		Menu	Check	Analysis
COMMENTS (*)				
Command		X	X	X
Description	Allows entering user-defined comments identifying task or defining variable names to the programmer.			
Syntax	* <Text>			
	Syntax Element	Description		
	<Text>	Any comment		
Comments	The (*) character must be the first non-blank character on a line to be recognized as comment mark. In other words, comments must be on separate lines from commands. Comments of more than one line must have the comment symbol at the beginning of each line.			
Examples	*End of Menu *ToolTips are included with the Buttons			
Program-Specific Features	In all programs the “*” should be placed in the first column. For comments longer than one line, the * character should be added for each new line			

		Menu	Check	Analysis
DEFINE				
Command		X	X	X
Description	DEFINE allows creation of new variables			
Syntax	DEFINE <Variable> [Scope]			
	Syntax Element	Description		
	<Variable> [Scope]	The name of the variable to be created Global/Permanent/Standard		
Comments	Variables in Epi Info 2000 do not have predefined types; they will accept any type of data that is assigned with the assignment command (“=”). However, once assigned, their type cannot be changed. <u>Scope</u> <u>PERMANENT</u> variables are stored in the EPIINFO.INI file or system			

registry and retain any value assigned until the value is changed by another assignment or the variable is UNDEFINEd. They are shared among Epi Info programs and persist even if the computer is shut down. Permanent variables in *Analysis* may not have values that depend directly or indirectly on table fields

GLOBAL variables persist for the duration of program execution. They are used in *Enter* to pass values from one view to a related view. They are used in *Analysis* to store values between changes of data source. Global variables in *Analysis* may not depend directly or indirectly on table fields.

STANDARD defined variables are used as temporary variables behaving like variables in the database. They lose their values and definitions at the next READ statement.

Examples

```
DEFINE Var1
DEFINE Var2 GLOBAL
DEFINE Var2 PERMANENT
```

Program-Specific
Features

Menu

The ASSIGN command in menu command blocks automatically defines a variable if it does not already exist. All variables created by the ASSIGN command in the menu are permanent variables.

Check Code

The DEFINE command can create standard, GLOBAL, or PERMANENT variables. Standard variables retain their value only within (all pages of) the current view and are reset when a new record is loaded. GLOBAL variables retain values across related Views and even when a new View is opened by the program, but are removed when the *Enter* program is closed.

Analysis

Standard Variables—those not defined as GLOBAL or PERMANENT—exist only until the next READ command. GLOBAL variables remain in effect as long as *Analysis* is running and can be used to carry values from one PGM to another. If a permanent variable is created and no value is assigned, it will be deleted before closing *Analysis*.

	Menu	Check	Analysis
<i>DIALOG</i>			
Command	X	X	X

Description The Dialog command provides interaction with the user from within a program. Dialogs can display information, ask for and receive input, and offer lists for making choices.

Syntax (Each command is on a single line. Note the differences between programs and pay close attention to commas and semicolons.)

Menu

- **DIALOG** "<Text Prompt>", {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>", <Variable>, **FORMAT**= "<FormatPattern>", {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>", <Variable>, **LIST**= "<Value1>"; "<Value2>"; "<Value3>"; ... ;"<Value_n>" {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>", <Variable>, {**BUTTONS**="value1";value2;etc}

Check code

- **DIALOG** "<Text Prompt>" {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>" <Variable> [<Entry type>] {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>" <Variable> "<Value1>", "<Value2>", "<Value3>", ... ,"<Value_n>" {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>" <Variable> [<List Type>] {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>" <Variable> **DBVALUES** <Table Name> <Field Name> {**TITLETEXT**="<Title>"}

Analysis

- **DIALOG** "<Text Prompt>" {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>" <Variable> [<Entry type>] {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>" <Variable> "<Value1>", "<Value2>", "<Value3>", ... ,"<Value_n>" {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>" <Variable> [<List Type>] {**TITLETEXT**="<Title>"}
- **DIALOG** "<Text Prompt>" <Variable> **DBVALUES** <Table Name> <Field Name> {**TITLETEXT**="<Title>"}

Syntax Element	Description
<Text Prompt>	Text to be displayed as message
<Variable>	Variable to store value entered.

[<Entry Type>]	A reserved word that defines the type of input to be taken and stored in the variable. If no type is specified, the value is number/date. TEXTINPUT will define variable as text YN will define variable as Boolean.
FormatPattern	A text string, such as "####" or ##.### that indicates the type of data to be entered. # indicates a digit.
<Title>	Text to be used as the caption for the dialog window.
[<List Type>]	DATABASES DBVIEWS

Comments

The first form of DIALOG places a dialog box on the screen, using the text provided, with an "OK" button.

The second form of DIALOG displays the text prompt and provides a means for entering a value. If YN is specified, "Yes," "No" and "Cancel" buttons are presented and the specified variable is set to (+), (-), or (.).

If TEXTINPUT or no entry type is specified, an entry field for user response is provided, with OK and Cancel buttons. The variable is assigned the value of the user's input, with a missing value if Cancel is chosen.

The third form of DIALOG displays the text prompt and provides a combo box for selecting among the values. OK and Cancel buttons are presented. The variable is assigned the value of the user's input, with a missing value if Cancel is chosen. The variable will be of text type.

The fourth form of DIALOG displays a combo box of databases or view variables. OK and Cancel buttons are presented. The variable is assigned the value of the user's input, with a missing value if Cancel is chosen. The variable will be of text type.

The fifth form of DIALOG displays a combo box of distinct values of the specified variable in the specified database. OK and Cancel buttons are presented. The variable is assigned the value of the user's input, with a missing value if Cancel is chosen. The variable will be of the same type as the database variable.

Examples

```
DIALOG "Enter Value " var1 TITLETEXT="Dialog 1"
DIALOG " Enter Value " var1 DATABASES
DIALOG " Enter Value " var1 DATABASES TITLETEXT="
Dialog 1"
DIALOG " Enter Value " var1 "aaa", "bbb", "ccc", "ddd"
TITLETEXT=" Dialog 1"
```

Program-Specific
Features

DIALOG "please select database" var1 DATABASES
TITLETEXT="database"

Menu Check Analysis

ENDBEFORE

Command

X

Description

ENDBEFORE divides Check commands to be executed before data entry from those executed after entry.

Syntax

ENDBEFORE

Comments

These commands will control actions so that they occur before or after accessing a View, record, page, or field.

The default time for execution of commands associated with a variable is "AFTER ENTRY." If execution is to occur before entry, place the relevant commands in a block called BEFORE ENTRY ... END within the block of commands for that variable. ENDBEFORE must be placed starting in the first position of the line. No other command can be placed in the same line.

If end before is not present, Epi Info 2000 will execute all commands after the condition.

ENDBEFORE can be used in pages, views, entries, and records.

Examples

The following commands represent the Check code for a variable called Demo1

Dialog 'This is Before entry' TITLETEXT = "Dialog1"
ENDBEFORE
Dialog 'This is After entry' TITLETEXT = "Dialog 2"

Program-Specific
Features

ENDBEFORE is only used in Check code. Command buttons will not take Endbefore since all code inserted will be executed when the user clicks on the button.

	Menu	Check	Analysis
EXECUTE(LINKTO OR ACTIVATE)			
Command	X	X	X

Description Executes a Windows or DOS program--either one explicitly named in the command or one designated within the Windows registry as appropriate for a document with the file extension that is named. This provides a mechanism for bringing up whatever word processor or browser is the default on a computer without first knowing its name.

- Syntax**
- EXECUTE ExecutableFile.EXE
 - EXECUTE Filename.FileExtension

Syntax Element	DESCRIPTION
<ProgramName>	Path and program name for EXE and COM files, along with any command line arguments desired. Filename for programs registered in windows. An Internet address (URL).

Comments

If the name of an executable program, such as ENTER.EXE or MYBATCH.BAT or even DOIT.COM is given, the program will be run in a separate window. When the program terminates, the window will be closed.

If the name given is not a program, but a file with an extension (the 3 characters after the ".") registered by Windows for displaying the document, the correct program to display the file will be activated. For example, WRITEUP.DOC might cause Microsoft Word[®] to run and load the file on one computer. On another machine, however, .DOC might correspond to Corel WordPerfect[®]. Usually .TXT will run NOTEPAD.EXE[®] or WORDPAD.EXE[®], and image files will appear either in a browser or in a graphics program on your computer. An .HTM file will bring up the default browser on your computer.

The exciting thing about this form of EXECUTE is that you do not need to supply the location or even the name of the program that will be run. These details are stored in the Windows registry for common file extensions. It is as though you said, "I want to drive a nail," and the operating system hands you a hammer. You do not have to search for the hammer and then the nail.

The same concept applies to Internet addresses (URLs). Give a URL (Universal Resource Locator) that ends in .HTM and Windows brings up the default browser on your computer, connects to the Internet (if possible), and goes directly to the site indicated. It is the beginning of “the World as your disk drive.” With this device and an Internet connection, you could set up a menu with buttons for websites in neighboring countries or states, favorite search engines, or whatever you need for daily work, thus making contact with these information sources just a mouse click away.

Examples
Program-Specific
Features

Execute “C:\epi2000\Epimap.exe”

Analysis

Link and Activate are not valid command names.

Check Command

In Check code, the EXECUTE command can be placed in any command block, but is often used with a button. A button does not have a Before Entry section.

Menu

Alternative syntax (Link, Activate) is available only for the menu.

		Menu	Check	Analysis
EXIT (QUIT)				
Command		X		X
Description	Closes current data files and terminates the current program, closing <i>Analysis</i> .			
Syntax	EXIT			
Comments	EXIT and QUIT have the same function			
Examples	<pre>DIALOG "Do you wish to close Analysis" result YN TITLETEXT="Close" IF result = (+) then Quit End</pre>			
Program-Specific Features	<u>Menu</u>			
	Closes the current menu. EXIT does not necessarily close programs that have been started by the menu in other windows. The user must			

close them, if this is desired.

Analysis

Exit will stop the execution of a program and close *Analysis*. To stop a program running in *Analysis* press the Ctr-Break keys. *Analysis* will close and the program will be stored as LASTPGM.PGM in the default directory

Menu Check Analysis

FREQUENCIES

Command X

Description **FREQ** produces a table from the table(s) specified in the last READ statement, showing how many records have each value of the variable. Confidence limits for each proportion are included.

Syntax **FREQ** <Variable>
FREQ * [EXCEPT <Variable>]

Syntax Element	Description
<Variable>	Donor variable (where the values are)

Comments Records may be included or excluded from the count by using SELECT statements. Those marked as deleted in *Enter* will be handled according to the current setting for SET PROCESS. If more than one variable name is given, FREQ will make a separate table for each variable.

Confidence limits for the binomial proportions are produced. Further details on the confidence limits are given in the chapter on Statistics.

If a WEIGHTVAR is specified, the value of the WEIGHTVAR variable is treated as a count of instances of the variable being FREQed. For example, in the following command a record containing the value 30 for AGE and 15 for COUNT would give a result equivalent to 15 individuals of age 30:

FREQ AGE WEIGHTVAR = COUNT

If STRATUM is specified, a separate frequency will be produced for each value of the stratifying variable.

FREQ ILL STRATAVAR=SEX, for example, will produce a table showing ILL (perhaps Yes/No/Unknown) for Males and another for Females. Note that the same numbers can be obtained using TABLES ILL SEX, but the latter will give the results in one table rather than in separate tables, and produce statistics to test for an association between

ILL and SEX.

FREQ * will make a table for each variable in the current view other than unique identifiers. It is often used to begin analysis of a new data set. To do frequencies of all variables except a few, use

FREQ * EXCEPT VarName(s) followed by the names of the variables to be excluded.

Examples

FREQ age
 FREQ AGE SEX RACE (Does separate frequencies for each of the variables)
 FREQ * EXCEPT NAME (Does frequencies of all variables except NAME)
 FREQ AGEGROUP STRATAVAR = COUNTY (Does a frequency of AGEGROUP for each COUNTY)
 FREQ AGEGROUP WEIGHTVAR = COUNT (Processes summary data containing a COUNT in each record, so that each record is considered to represent the number of individuals contained in its COUNT field. This replaces the SUMFREQ command in Version 6 of Epi Info.)

Menu Check Analysis

GOTO

X

Command

Description

GOTO can be used alone or as a consequence of an IF statement to transfer the cursor to a particular named variable field. The special uses are:

Syntax

GOTO <Event>

Comments

Syntax Element	Description
<Event>	Can be a variable, PageEnd, RecordEnd, or NextRec
+1	Automatically saves the current page, if changes have been made, and goes to the next page.
-1	Automatically saves the current page, if changes have been made, and goes to the previous page.
<Page Number>	Automatically saves the current page, if changes have been made, and goes to the page indicated by the number.

Examples
Program-Specific
Features

Goto +1

Check Command

Unlike the JUMPS command in Epi Info Version 6, the GOTO command will make all variables in between unavailable for data entry (Read Only).

Menu Check Analysis

GRAPH

Command

X

Description

Numerous settings are available in the GRAPH module, and these can be saved as a Graph Template. When a Graph Template is referred to by name, the settings are taken from this template. If explicit settings are given as above, they override the settings in the Graph Template.

Syntax

GRAPH <VarName(s)> [* <Varname>] **GRAPHTYPE=** "<GraphType>" {**TITLETEXT**="<Text>"}
 {**WEIGHTVAR**=<Variable>} {**TEMPLATE**='<Filename>'}
 {**WINMETAFILE**='<Filename>'} {**YTITLE**="<Text>"}
 {**PERCENTS**=(+)} {**XTITLE**="<Text>"} {**XRANGE**= <low> **TO** <High>} {**YRANGE**= <Low> **TO** <High>} {**XTICK**= <Xnumber>}
 {**YTICK**=<Ynumber>} {**THREED**= (?)}
 {**XLABELS**="<Text>","...",<Text>"}
 {**LEGEND**="<Text>","...",<Text>"}
 {**CLIPBOARD**=(?)} {**XORIENT**="Vertical"|"Horizontal"}

GRAPH [VarName(s)] <TemplateFile>

Syntax Element	Description
<GraphType>	HISTOGRAM BAR SCATTER LINE PIE HORIZONTAL BAR SPLINE MARK
	AREA PARETO HI-LOW SURFACE POLAR CUBE DOUGHNUT
<Low>	Represents the lower limit to be plotted in the graph for that specific axis
<High>	Represents the upper limit to be plotted in the graph for that specific axis
(?)	(+) or ON to apply the specified option; (-) or OFF to not apply it.

Comments

WEIGHTVAR is intended to be used for summary data in which one

variable contains the frequency for the event.

Since creating graphs might be tedious, all properties for a specific graph can be stored in a **TEMPLATE**, which will be identified with the .CHT extension. If the graph will be used by any other application, it can be saved as a **Windows Metafile** image using the **WINMETAFILE** setting or on the clipboard using the **CLIPBOARD** setting.

Selecting **PERCENTS=(+)** can plot proportions. Y Axis titles and scale can be defined using **YTITLE**, **YRANGE** and **YTICK**. **LEGEND** can be used to specify labels for the legend; if not specified, either the variable name or prompt will be used, depending on the **PROMPTS** setting.

X Axis titles and the scale can be defined using **XTITLE**, **XRANGE**, and **XTICK**. **XLABELS** can be used to specify labels for each value of the X axis; if not specified, the value will be used. **XORIENT** can be used to specify the direction in which the X axis labels will be displayed.

THREED= (+) means "three dimensional"

Examples

```
GRAPH BAKEDHAM CABBAGESAL * AGE
.GRAPHTYPE="Bar" TITLETEXT="titulo" WEIGHTVAR=SEX
TEMPLATE='temp' BITMAP='bit' YTITLE="Count"
PERCENTS=(+) XTITLE="label2"
```

Menu Check Analysis

HEADER

Command X

Description Set up specific headings as part of the output in *Analysis*

Syntax **HEADER** <Level> ["<Text>" [(<FontProperty>)] [**APPEND**]]
 [**TEXTFONT** <Color> <Size>]
HEADER <Level> **DROP|RESET**

Syntax Element	Description
<Level>	Body Text Procedure Title Window Title Variable Title File Title Stratum Title Data Source Title
<Text>	Type text to be displayed
<FontProperty>	Properties (Underline, Bold, Italic) will be separated by commas (,)

<Color>	Aqua Black Blue Fuchsia Gray Green #xxxxxx, where x is a hexadecimal digit; the pairs of digits represent the amount of red, green and blue, respectively, on a scale of 0-255, included in the color.	Lime Maroon Navy Olive Purple	Red Silver Teal White Yellow
<Size>	Any number in the following ranges: From -7 to -1, from 1 to 7, and from +1 to +7		

Comments

Values for font size represent standard values for HTM font sizes. Numbers preceded by + or - increase or decrease the default size by the specified number of levels.

In the first format, use of the APPEND keyword will result in the specified title line being added to any title already defined for that level.

In the second format, use of the DROP keyword will result in the last title line for that level being removed. Use of the RESET keyword will result in the title line(s) and font characteristics for that level being reset to their Epi Info defaults (in most cases, blank).

Examples

```

READ 'D:\EPI2000\Sample.Mdb':viewOswego
HEADER 0 "This is HEADER 0" (BOLD) TEXTFONT Green 7
HEADER 1 "This is HEADER 1" (ITALIC) TEXTFONT Yellow 6
HEADER 2 "This is HEADER 2" (UNDERLINE) TEXTFONT Blue 5
HEADER 3 "This is HEADER 3" (BOLD, ITALIC,) TEXTFONT Red 4
HEADER 4 "This is HEADER 4" (BOLD, ITALIC, UNDERLINE)
TEXTFONT Teal 3
HEADER 5 "This is HEADER 5" (BOLD, ITALIC, UNDERLINE)
TEXTFONT Olive 2
HEADER 6 "This is HEADER 6" (BOLD, ITALIC, UNDERLINE)
TEXTFONT White 1
FREQ ILL
TABLES VANILLA ILL

TABLES VANILLA ILL STRATAVAR= SEX
    
```

HELP

Command

Menu	Check	Analysis
X	X	X

Description

Displays a customized document in HTM format using the default browser provided with Epi Info 2000

Syntax **HELP** <HTMLFileName{ #BlockName}>

Syntax Element	Description
<FileName>	Helpfile

Comments HTMLFileName must end in .HTM or .HTML. If an optional BlockName is given, the browser searches for named anchors (“bookmarks”) in the HTML file and positions the anchor at the top of the screen. An anchor in HTML can be created by using a tag like .

If no directory path is given, the HELP command will look first in the HELP directory under the currently designated language directory (e.g., .\SPANISH\HELP. If the file is not found, it looks in the default HELP directory, .\ENGLISH\HELP. That failing, it looks in the currently logged directory. This allows translated help files to be accessed automatically as long as they retain the name (and block names) of the corresponding English versions.

Filenames including spaces must be enclosed in single quotes. Anchors containing spaces must be enclosed in double quotes.

Examples
 HELP 'South Pacific'.HTM#"Bali Hi"
 HELP Function.htm#YEAR

Program-Specific Features

Menu Check Analysis

HIDE, UNHIDE

Command X

Description HIDE hides a field from view and makes it NOENTER.
 UNHIDE makes a field visible and returns it to the status it had before it was hidden.

Syntax **HIDE** <Field name(s)>}
UNHIDE <Field name(s)>

Syntax Element	Description
<Field name(s)>	One or more valid variables

Comments If no field name is specified, the current field (the one to which the


```

END
IF VOMITING = (+) THEN
  ASSIGN MAJORSYMP = MAJORSYMP + 1
END
IF MAJORSYMP >= 2 THEN
  ASSIGN CASE = (+)
ELSE
  ASSIGN CASE = (-)
END

```

(If two or more symptoms are present, the record is classified as a case.)

When the IF conditions do not describe all possibilities, standard defined variables will be reset after each record to the “missing” value. It is important to cover all the conditions in IF statements, to avoid gaps in the logic and results.

Sometimes it is important to have an “ELSE” condition that covers conditions not included in other IF clauses. This effect can be achieved by setting the variable initially to something other than missing, as in:

```

DEFINE ILL
ASSIGN ILL = (-)
IF VOMITING = (+) THEN
  ASSIGN ILL = (+)
END
IF DIARRHEA = (+) THEN
  ASSIGN ILL = (+)
END
IF FEVER = (+) THEN
  ASSIGN ILL = (+)
END

```

Sets Ill = (+) only if one or more symptoms are present.

The same result would be achieved by:

```

IF (DIARRHEA = (+)) OR (VOMITING = (+)) OR \
  (FEVER = (+)) THEN
  ASSIGN ILL = (+)
ELSE
  ASSIGN ILL = (-)
END

```

(Note that “\” is used to indicate continuation of a long line.)

An “expression” is a combination of variables, literal values, and operators that can be evaluated to produce a single result. Expressions can be quite complex.

Examples IF (CHEESE = (+)) AND NOT (PEPPERONI = (+)) THEN
 ASSIGN VEGI = (+)
 END
 IF (AGE <30) AND (AGE > 17) THEN
 ASSIGN GROUP = "Young Adult"
 END

Menu Check Analysis

KMSURVIVAL

Command X

Description Performs Kaplan-Meier Survival Analysis and graphs and statistics for one or several groups of subjects being followed in a clinical study. At any given time, some of the subjects may be "censored," that is, not have information available on their status. KMSurvival is especially constructed to deal with this situation.

Syntax **KMSURVIVAL** <Exposure> = <Outcome> * <CensorVar>
 (<Value>) **TITLETEXT**="<Text>" **TIMEUNIT**="<TimeUnit>"
 TEMPLATE='<Filename>'

Syntax Element	Description
<Exposure>	Variable in database
<Outcome>	Variable in database
<CensorVar>	Variable in database
<Value>	Value for uncensored variables
<TimeUnit>	Days Weeks Hours Years Months
<Filename>	A valid path and filename

Comments The KMWin program can also be run from the main menu of Epi Info. In this mode, however, it reads only Epi Info 6 (DOS) files. When run from *Analysis*, *Analysis* generates the necessary Epi Info 6 file automatically and runs the program after transmitting the parameters to set up the model. Once in KMWin, the model can be modified using the menu commands in KMWin.

Examples

Menu Check Analysis

LIST

Command

X

Description

LIST does a line listing of the current dataset. If variable names are given, LIST will list only these variables. LIST * will list all variables of all active records, using several pages across to accommodate all the variables, if necessary

Syntax

- **LIST** [*** EXCEPT**] <VarName(s)> {**COLUMNSIZE**= <Num1>, <Num2>, <Num3>} {**NOIMAGE**} {**NOWRAP**} {**LINENUMBERS**}
- **LIST** [*** EXCEPT**] <VarName(s)> [**GRIDTABLE/UPDATE**]

Syntax Element	Description
*	Used to represent all variables
<VarName(s)>	One or more variable names
<Num1>	A number from 0 to 99 representing the number of columns to be displayed before splitting a table (default=6). If zero, all specified columns will be displayed without splitting the table.
<Num2>	A number from 0 to 99 representing the number of columns to be allocated per image (default=0 for rendered images, 3 for unrendered images). If zero, each image column will allocated space based on its width in the view.
<Num3>	A number from 0 to 99 representing the number of columns to be allocated per memo field (default=0). If zero, each memo column will be displayed separately.
NOIMAGE	If present, image filenames are displayed, but the image itself is not rendered.
NOWRAP	If present, no line breaks are inserted into text fields, so that columns are as wide as the longest line in the field.
LINENUMBERS	Displays line numbers when automatic splitting is not being used.
GRIDTABLE	Data is displayed as grid instead of HTM format for viewing only.
UPDATE	Data is displayed as grid instead of HTM format for updating. Allows permanent changes in the database.

Comments

Changes made to the database using UPDATE do not use Check code or the properties set in *MakeView* (for example, must enter or no enter). If Check code features are desired, use *Enter* to edit data if the Update is convenient for changing a series of records in which one variable may have errors, for example.

Adding an EXCEPT Variable list will exclude all the named variables from a LIST or LIST *. This is convenient if you wish to leave out names or addresses, for example. Record numbers are listed automatically as the first item on each line when the program determines that a table must be split; when splitting is disabled, line numbers may be included by use of the LINENUMBERS keyword.

If the dataset has been sorted with the SORT command, the records are listed in sorted order. Otherwise, they are listed in an order determined by the Jet database.

If UPDATE is used, the data in the grid can be edited. Edits take effect when the cursor is moved to a new table row or when the grid is closed.

Examples

```
LIST
LIST name age sex race
LIST *
LIST * EXCEPT NAME SSN ADDRESS CITY STA
```

Menu Check Analysis

LOGISTIC

Command

X

Description

Performs a logistic regression analysis

Syntax

LOGISTIC <Outcome>= <VarName(s)> {**WEIGHTVAR** = <Variable>} {**TITLETEXT** = <text>} {**TEMPLATE** = <filename>}

Syntax Element	Description
<Outcome>	One variable in the database to be considered outcome. This must be a binary (2-value) variable or a Yes/No variable.
<VarName(s)>	One or more variables to be included in the model
<text>	Title of the project
<Variable>	Variable containing the frequency for the event.

Comments

The outcome variable must be numeric and contain values 0 and 1 only.

Examples

Program-Specific
Features

Menu Check Analysis

MAP

Command

X

Description

Numerous settings are available in the MAP module. These can be saved into a MAP template. When a MAP template is referred to by name, the settings are taken from this record. If explicit settings are given (as above), they override the settings in the MAP Template

Syntax

MAP <Aggregate Function>(<VarName>) <GeoVarName> ::
'<Shapefile>':<Variable> {**TITLETEXT** = "<text>"} {**TEMPLATE**
= '<filename>'} {**DENOMINATOR** = <var2>} {**WEIGHTVAR** =
<Var3> }

Syntax Element	Description
Aggregate Function	SUM COUNT MAXIMUM AVERAGE MINIMUM
<VarName>	Represents the variable in the dataset that contains information about the geographic region.
<GeoVarName>	Represents the variable in the dataset that identifies the geographic region
<Shapefile>	Path and file name for the Shapefile (SHP) to be used. Shapefiles are those used by ArcView and Arc/Info.
<Variable>	Variable in the SHP file that corresponds to the GeoVarName in the dataset.
<var2>	Variable in the SHP file that contains the denominator for population- or area-based rates.
<Var3>	Variable containing the frequency for the event.

Comments

All rates are given using the denominator units; to create other rates (such as per 1000 population), define a new variable that stores the desired rate (1000, etc) and use it as WEIGHTVAR.

Values to be used as denominator should be included in the DBF file associated to the shape map.

Examples

MAP SUM(DENGUE) ADMIN_NAME ::
'D:\EPI2000\Venezuela\venezuel.shp':ADMIN_NAME

Menu Check Analysis

MATCH

Command

X

Description

MATCH performs a matched analysis of the specified exposure and outcome variables, which are assumed to be yes/no variables. One table is produced for each number of cases in a match group. The first variable will appear on the left margin and will contain values from zero to the number of cases in the match group. The second variable will appear on the top margin and will contain values from zero to the number of cases in the match group. The cells contain the number of match groups showing the combination of positive exposures and positive outcomes shown in the margins. The output table produced by the command is similar than that produced by TABLES.

Syntax

MATCH <Exposure> <Outcome> **MATCHVAR=** <Variables>
{WEIGHTVAR= <Variable> **}** **{OUTTABLE =** <Tablename> **}**
{COLUMN SIZE = <Number> **}** **{NOWRAP}**

Syntax Element	Description
<Exposure>	Variable in the database to be considered the risk factor
<Outcome>	Variable in the database considered Disease of consequence
<Variables>	Variable(s) used to identify matched pairs
<Variable>	Variable containing the frequency for the event.
<Tablename>	Name for a table to store summarized data
<Number>	Number of columns per line of output; zero means do not split lines
NOWRAP	If present, no line breaks are inserted into text fields, so that columns are as wide as the longest line in the field.

Comments

The **MATCHVAR** variable is used for stratification. For example, if every case has a corresponding control, then each pair is given an identifier, perhaps PAIRID, that identifies the pair. The **MATCHVAR** variable is PAIRID and the **MATCH** command internally performs a Mantel-Haenszel stratified analysis using PAIRID as the stratifier. Matched groups can have several controls per case, or even more than one case matched to several controls.

WEIGHTVAR is intended for summary data in which the number of cases will be stored in a variable such as count.

Examples

In a review of hospital records of patients with and without postoperative infection, a single case is matched by age, sex, and operation type to as many controls (1 to 4) as are available. Each case record and the matched control record(s) is given a **GROUPID** number.

The INFECTION variable indicates the presence or absence of postoperative infection. The analysis of participation of Dr. A. in the operation is done as follows:

MATCH INFECTION DRA GROUPID

Menu Check Analysis

MEANS

Command

X

Description

Syntax

MEANS <Variable1> [<Variable2>] {**STRATAVAR=**
<variable(s)>} {**WEIGHTVAR=** <variable>} {**OUTTABLE =**
<Tablename>}

Syntax Element	Description
<Variable1>	A numeric variable to be used to calculate means (or * for all numeric variables)
<Variable2>	Any variable used for cross-tabulation (optional, or * for all numeric variables)
<Variable(s)>	Variable(s) to be used for stratified analysis
<Variable>	Variable containing the frequency for the event.
<Tablename>	Name for a table to be created

Comments

The MEANS command has two formats. If only one variable is supplied, the program produces a table like that produced by FREQUENCIES, plus descriptive statistics.

If two variables are supplied, the first is a numeric variable containing data to be analyzed and the second is a variable that indicates how groups will be distinguished. The output of this format is a table like that produced by TABLES, plus descriptive statistics of the numeric variable for each value of the group variable.

MEANS produces the following statistical tests:

Parametric tests

ANOVA (for two or more samples)

Student's t-test (for two samples)

Non-parametric tests

Kruskal-Wallis one-way analysis of variance (for two or more samples)

Mann-Whitney U Test = Wilcoxon Rank Sum Test (for two samples)

Further details are given in the chapter on Statistics.

Examples

MEANS AGE ILL

Will compare ages for those in whom ILL = (+) with those for whom ILL = (-). If there are more than two groups in the variable ILL, such as 1, 2, and 3, or if SET IGNORE = OFF has been invoked, so that (+), (-), and (.) are all considered groups, MEANS will do a comparison of all the groups and the answer will indicate whether or not the groups differ from each other (but not in what pattern).

MENU

Command

Menu Check Analysis

X

Description
Syntax

MENU <FileName>.MNU {<Image>}{<Left>, <Top>},
{<Width>, <Height>}

Syntax Element	Description
<FileName>	
<Image>	Image to be used as background for the menu. Epi2000.exe supports GIF, JPEG, and BITMAP images.
<Left>	A number from 1 to 99 representing the distance from the left side of the window in percentage points.
<Top>	A number from 1 to 99 representing the distance from the top border of the window in percentage points.
<Width>	A number from 1 to 99 representing the width of the image desired.
<Height>	A number from 1 to 99 representing the distance from the top border of the window in percentage points.

Comments

The MENU command loads another menu with the given file name from within a block of commands in an MNU file. If the name of an image file is given (either .BMP or .BID), the image is loaded on the main form. Left and Top locations can be given to position the upper left corner of the menu on the screen. They are given as percents of screen width and height: 0, 0 is the upper left corner of the screen. Width and Height are also percentages of the screen. If no locations and sizes are given, those of the main menu are used.

Examples MENU SURVEIL.MNU REFCAMP.BMP 5,5, 50, 50

It displays the surveillance menu, an image of a refugee camp from the air, and places the menu 5% right and 5% down from the upper left corner of the screen, extending half the screen width and half the screen height so that it occupies one quarter of the screen.

MENUITEM Menu Check Analysis
 Command X

Description MENUITEM is used to create a non- hierarchical option in the vertical menu

Syntax **MENUITEM** "<Text>", <Block>

Syntax Element	Description
<Text>	Text to be displayed
<Block>	Block to be executed (see COMMAND BLOCK)

Comments Every Menu in Epi Info 2000 contains two main components. The first part defines the menu items, buttons, and screentext, and the second part defines the command blocks. The example below shows a single pulldown menu definition.

```

Examples
BEGIN
  POPUP "&Programs "
  BEGIN
    MENUITEM "Ma&ke View (Questionnaire)" , MakeView
    MENUITEM "&Enter Data", Enter
    MENUITEM "&Analyze Data", Analysis
    MENUITEM "&StatCalc", StatCalc
    MENUITEM "Epi Ma&p", EpiMap
    MENUITEM "&Nutrition", Nutstat
    MENUITEM "&Visualize Data", VisualizeData
    MENUITEM "&Word Processing", Eped
    MENUITEM "E&xit ", Exit
  END
END
    
```

Menu Check Analysis

MERGE

Command

X

Description

Merges records in one dataset with those in another, using one or more defined identifiers to establish the match between records. Records in the second dataset can be appended to the end of a dataset, used to update records in the main dataset, or joined to those in the first dataset “side-by-side.”

Syntax

MERGE <Table Specification> <MergeType>
{LINKNAME= <Text> **}** <Key(s)>

FILESPEC

Syntax Element	Description
<Table Specification> <MergeType>	The type and name of a data table to be read as the Merge file. See Read. If no type is specified, the effect is the same as both APPEND and UPDATE below: matching records are updated with new material that may appear in the merging file, and those that do not match are added to the end of the main file.
Key(s)	One or more expressions that designate keys on which the match or relate will be performed. These are in the form: <ExpressionCurrentTable>::<ExpressionMergeTable>
FILESPEC	A special set of instructions for accessing files other than Access files, often those with password security set. See READ for more details.

Comments

APPEND adds unmatched records in the Merge table to the currently active dataset. Only fields found in both datasets will be added.

UPDATE will replace fields of records in the active table with those in the Merge table if the key expressions match. Only fields found in both datasets with a non-empty value in the Merge table will be replaced.

RELATE moves the unique key of the current table to the foreign key of the related table to make a permanent relationship. The related (Merge) table must be an Access/Epi2000 table. If it started life as an Epi Info 6 file, it must be imported using the FILE|IMPORT feature, which will insert the necessary fields (FKEY and others) in the resulting Access/Epi2000 table.

Key(s) In most cases the expressions are the names of variables in the tables that will serve as keys for the match. They can also be

mathematical expressions including variable names that would produce a unique match on particular records. More than one key pair (“multiple keys”) can be designated, separated by AND.

See READ for more information about database formats and FileSpecs.

Examples

Program-Specific
Features

	Menu	Check	Analysis
MUSTENTER			
Property		X	
Description	MUSTENTER specifies that missing values are not allowed. If <Enter> is pressed in a MUSTENTER field before a specific entry has been made, the cursor remains in the field until an entry is made.		
Comments	MUSTENTER cannot be used in combination with READONLY since those properties are mutually exclusive.		

	Menu	Check	Analysis
PICTURE			
Command	X		
Description	Loads the specified image, but does not store the image name between sessions.		
Syntax	PICTURE <Image>		
	Syntax Element	Description	
	<Image>	Image to be displayed. Epi Info 2000 supports BMP, GIF, and JPEG images.	
Comments	Picture can be added in a command block. In that case, the background image will change when the command is executed.		

LEGAL values. The missing value is accepted unless the field is also designated MUSTENTER. Ranges may be numeric, dates, or strings (text).

Comments

Range can be used only for numeric variables. If a range is required for dates or times, it is necessary to use IF statements in the check code.

Menu Check Analysis

READ

Command
Description

X

READ makes one or more views the active dataset. It also removes any previously active datasets and associated DEFINEd variables, and dataset-specific commands.

Syntax

READ <Table specification> **LINKNAME=** <LinkTable Name>
FILESPEC HDR="NO" FMT="<File format>**"**

Syntax Element

DESCRIPTION

<Table Specification> LinkTable Name	See the Comments section below
FILESPEC	Additional information necessary to process the data table, which depends upon the type of data being processed. See Comments below.
"NO" <File Format>	

Comments

READ erases or "forgets" all previous IF, SELECT, ASSIGN, SORT, and RECODE statements. All DEFINEd variables except for those designated GLOBAL or PERMANENT are removed. READ can therefore be used to reset SELECT, IF, DEFINE, and SORT statements previously entered, and begin processing with a clean slate (except for

GLOBAL and PERMANENT variables, which will retain their definitions and values).

If the file to be READ is not in Microsoft Access format, a LINK to the external files will be placed in the current MDB. A link is a small Access table giving information about the external file. If no LinkTableName is specified, a temporary table will be created. After a link is created, it can be treated like an internal Access/Epi2000 table, although the data remain in the original file.

Using Various File Types

The READ, RELATE and MERGE commands can operate on many different types of data. Each type requires a different table specification, and some types required additional information in a file specification. Table specifications usually consist of a data type (contained in double quotes), a space, a file path (which may be enclosed in single quotes, and must be if it includes a space), a colon, and a table name. File specifications must start on a new line with the word FILESPEC; this may be followed by keyword parameters in the form <keyword>=<value>; this may be followed by field definitions, each on a separate line, in the form <fieldname> <start>-<end> <fieldtype>; followed by END (on the same line if only keywords are present, on a new line if field definitions are present).

Epi2000 Views and Access Tables

If the view or table is in the current project, the first form may be used. The second form is like the first, but explicitly specifies the file path; it may be used with old-style (8.3) path and filenames. The third form specifies an Access database; single quotes must be used around the path because it contains a space. The fourth form shows a password-protected Access database (using the database password method, not the workgroup file method).

Desktop Databases

These databases have only one table per file, so there is no need to specify a table. Note that later versions of Paradox must be read using the ODBC driver supplied with professional versions of the product. Also note that the Epi6 driver actually reads the REC file into the database, so changes made through Epi6 after the READ will not be saved to Epi2000 and changes made through Epi2000 will not be saved to Epi6 (unless the Epi6 file is rewritten with the WRITE REPLACE command).

Microsoft Excel

Excel 3.0 had only one spreadsheet per file, so there is no need to specify a table if the entire spreadsheet is to be read, as shown in the

first example. The FILESPEC uses the HDR keyword to indicate whether the first row contains field names or not. The second example shows how to use a cell range with Excel 3. Later versions of Excel allow multiple spreadsheets per file; to refer to a particular spreadsheet, use its sheet name followed by a dollar sign, as shown in the third example. Alternatively, you may specify a named range, as shown in the fourth example. The fifth example shows how to specify a cell range in conjunction with a sheet name.

ODBC Databases

Data sources, which must be accessed via ODBC drivers, include SQL Server, Oracle, SAS and later versions of Paradox. ODBC drivers are not part of Epi Info 2000 and must be obtained from the database vendor or a third party provider. Once an ODBC driver is installed on a user's, data source names must usually be established for each database to be accessed using the driver; this is done through the ODBC Manager program in the Control Panel, and is usually done by the database administrator. The FILESPEC may be used to supply a username and password, but if none is supplied and the database is password-protected, the program will prompt the user to supply them.

Text Files

There are two basically different forms of text files. Both forms have only one table per file, so there is no need to specify a table. Both forms puts the data for one record on a single line. The difference is in how the fields are indicated. One form, called "delimited", uses designated characters to separate fields. The first example shows the CSV version, where commas separate fields and text fields are surrounded by double quotes. The second example shows the TAB version, where tab characters separate fields. The third example shows the DLM version, which explicitly includes the ASCII value of the delimiter character (in this case, space). The command generator allows the delimiter character to be set explicitly.

The second form, shown in the fourth example, is called "fixed", because each field occupies the same positions in each line. All character positions through the last field must be accounted for, even if they do not contain useful data (the command generator will automatically generate filler fields as required). Even if the first line of the file contains field names (HDR="YES"), the names specified in field definitions will be used.

Note that the text file driver actually reads the file into the database, so changes made to the file after the READ will not be saved to Epi2000 and changes made through Epi2000 will not be saved to the text file (unless it is rewritten with the WRITE REPLACE command, which is available only in CSV format).

HTML Files

An HTML file may contain more than one table, but since the tables do not have names, the table name portion of the specification is not used. Instead, the line and offset in the file at which the table starts is specified. The command generator examines the entire file and finds the locations of the tables, which it then displays for the user to preview and select; it is extremely difficult to determine the line and offset without using the command generator. Note that the HTML file driver actually reads the file into the database, so changes made to the file after the READ will not be saved to Epi2000 and changes made through Epi2000 will not be saved to the HTML file.

Examples

Epi2000 Files

```
READ viewOswego
READ "Epi2000" c:\Epi2000\sample.mdb:viewOswego
READ "Access 97" 'c:\My Files\My Data.mdb':MyTable
READ "Access 97" 'c:\My Files\Secret Data.mdb':MyTable
FILESPEC PWD="MyPassword" END
```

Desktop Databases

```
READ "Epi6" c:\Epi2000\dataepi6.rec
READ "dBase III" c:\Epi2000\datadb3.dbf
READ "dBase IV" c:\Epi2000\datadb4.dbf
READ "dBase 5" c:\Epi2000\datadb5.dbf
READ "Foxpro 2.0" c:\Epi2000\datafp20.dbf
READ "Foxpro 2.5" c:\Epi2000\datafp25.dbf
READ "Foxpro 2.6" c:\Epi2000\datafp26.dbf
READ "Paradox 3.x" c:\Epi2000\datapdx3.db
READ "Paradox 4.x" c:\Epi2000\datapdx4.db
```

Microsoft Excel

```
READ "Excel 3.0" c:\Epi2000\dataxl3.xls
FILESPEC HDR="YES" END
READ "Excel 3.0" c:\Epi2000\dataxl3.xls!a1:e74
FILESPEC HDR="NO" END
READ "Excel 4.0" c:\Epi2000\dataxl4.xls:Sheet1$
FILESPEC HDR="YES" END
READ "Excel 5.0" c:\Epi2000\dataxl5.xls:RangeName
FILESPEC HDR="NO" END
READ "Excel 8.0" c:\Epi2000\dataxl8.xls:Sheet1$a1:e74
FILESPEC HDR="NO" END
```

ODBC Databases

```
READ "ODBC" SASDataDSN:MyTable
READ "ODBC" SQLDataDSN:BigTable
FILESPEC UID="Username" PWD="Password" END
```

Text Files

```

READ "Text (Delimited)" c:\Epi2000\datacsv.txt
FILESPEC HDR="YES" FMT="CSV" END
READ "Text (Delimited)" c:\Epi2000\datatab.txt
FILESPEC HDR="NO" FMT="TAB" END
READ "Text (Delimited)" c:\Epi2000\datadlm.txt
FILESPEC HDR="YES" FMT="DLM(32)" END
READ "Text (Fixed)" c:\Epi2000\datafix.txt
FILESPEC HDR="NO" FMT="FIX"
      Field1 1-5 Text
      Second 6-12 Date
      Third 13-15 Number
      Field4 16 YesNo
END
    
```

HTM Files

```

READ "HTML" 'D:\analysis\OUT704.htm'
FILESPEC LINE="16" OFFSET="642" END
    
```

	Menu	Check	Analysis
READ ONLY			
Property (A checkbox in the field dialog)		X	
Description	Allows defining variables that contain calculated values.		
Comments	READ ONLY in a field keeps the user from placing the cursor in the field or entering data. It is particularly useful for calculated fields that are not to be changed directly. Read Only cannot be used with MUSTENTER.		

	Menu	Check	Analysis
REPEAT			
Property (a checkbox in the Field dialog)		X	
Description	Causes the value of the same field in the last record accessed to become the default value for the designated field in the current record.		

Comments Saves keystrokes during data entry for fields where the value seldom changes.

Examples A county or state name is often the same for record after record in a given locality. Making it a REPEAT field brings up the last value entered with each new record. In the few records in which the correct answer may be another state or county, the data entry person can easily change the value before the record is saved.

Menu Check Analysis

RECODE

Command X

Description
Syntax

RECODE <Variable1> **TO** <Variable2>
 Value1 - Value2 = "<Recoded Value>"
 Value2 - Value3 = "<Recoded Value>"
 Value3 = "<Recoded Value>"
 ELSE="<Recoded Value>"
END

Syntax Element	Description
<Variable1>	Donor variable (where the values are)
<Variable2>	Receiver Variable (where recoded values will be)

Comments Text must be enclosed in quotation marks. Numeric ranges are separated by a space, hyphen, and space, as in 1 - 5. Negative values are permitted, as in -10, -9 - -8. The words LOVALUE and HIVALUE may be used to indicate the smallest and largest values representable in the database. The word ELSE may be used to indicate all values not falling in the preceding ranges. Recodes take place in the order stated; if two ranges overlap, the first in order will apply.

Examples RECODE AGE TO var1
 LOVALUE - 0 = "<=0"
 0 - 10 = ">0 - 10"
 10 - 20 = ">10 - 20"
 20 - 30 = ">20 - 30"
 30 - 40 = ">30 - 40"
 40 - 50 = ">40 - 50"
 50 - 60 = ">50 - 60"

```
60 - 70 = ">60 - 70"
70 - 80 = ">70 - 80"
80 - 90 = ">80 - 90"
90 - 99 = ">90 - 99"
99 - HIVALUE = ">99"
```

END

Program-Specific
Features

Analysis

Analysis cannot RECODE more than about 12 levels of values. To bypass this limitation, there is an example in the How To Chapter about alternate methods of recoding.

REGRESS

Menu Check Analysis

Command X

Description This command performs multiple linear regression

Syntax **REGRESS** <Outcome> = <Predictor(s)>

Syntax Element	Description
<Outcome>	Variable containing result or effect
<Predictor(s)>	One or more numeric variables that can be used as predictor for a linear regression model.

Comments The outcome variable must be numeric or Yes/No filed
 The formula and an explanation of the results are given in Applied Regression Analysis, 2nd edition, by Draper and Smith (John Wiley & Sons, New York, 1981, p. 84 ff).

Examples

RELATE

Menu Check Analysis

Command X X

Description The RELATE command links one or more tables to the current dataset during analysis, using a common identifier to find matching records.

The identifier may span several fields, in which case values in each of the fields must match. The changes are not permanent, and the linked tables remain separate on the disk, having been linked only temporarily by a system of pointers in random access memory.

Syntax

RELATE <Table Specification>
{LINKNAME= <Text> **}** <Key(s)> [ALL]

FILESPEC

Syntax Element	Description
<Table Specification> Key(s)	The type and name of a data table to be read as the Related file. See Read. One or more expressions that designate keys on which the match or relate will be performed. These are in the form: <ExpressionCurrentTable>:: ExpressionMergeTable >
ALL	If present, records in the table(s) already READ or RELATED for which there is no corresponding record in the RELATE table will be included in the data with null values for variables in the RELATE table. If absent, only records with corresponding records in the RELATE table will be included in the data.

Comments

To use RELATE, at least one table must have been made active with the READ command. The table to be linked must have a key field such as HOUSEID that has identifiers relating records in the two tables. In Epi Info 2000, the keys in the main and related tables or files do not have to have the same name. Epi Info 2000 can also create relationships with more than one variable at the same time; the number of variables on both sides of the :: must be the same, however.

If the keys were created automatically by Epi Info 2000, they will both be called "UniqueID" and you do not have to specify keys. If you try to relate two tables with no automatic relationship established, *Analysis* will notify you that the relationship could not be completed.

In most cases the expressions are the names of variables in the tables that will serve as keys for the match. They can also be mathematical expressions including variable names that would produce a unique match on particular records. More than one key pair ("multiple keys") can be designated, separated by AND.

The related table may have fewer records than the main table, but at least some of the records should have the same identifier in the identifier field.

After issuing the RELATE command, the variables in the related table may be used as though they were part of the main table.

Frequencies, cross-tabulations, and other operations involving data in both the main and related tables can be performed. If a permanently linked table is desired, the WRITE command may be used to create a new, permanently linked table.

More than one table may be RELATED to the main table by using successive RELATE commands.

Sometimes a variable name may be duplicated in more than one table. In referring to a variable in a related table, you may (optionally) use the form: HOUSE.AGE to represent the variable AGE in the View HOUSE. This will distinguish it from another variable AGE that might be in the main table.

By using the relational features of *Enter* and Check code and the RELATE command in *Analysis*, you can set up relational (actually “hierarchical”) databases just as in dBASE or other database management systems. This provides powerful database handling in combination with the epidemiologic statistics available in *Analysis*.

Examples

```
READ 'D:\EPI2000\Refugee.MDB':viewFamily
RELATE viewPatient FAMIDNUM :: FAMIDNUM
RELATE viewAssmnt BOH :: BOH
```

Program-Specific
Features

Check Code

Relationships are created in Epi Info 2000 as a new type of variable. A relationship created in Epi6 will not be converted automatically into one in Epi Info 2000. Use [MERGE](#) to convert an Epi6 relationship into one in Epi Info 2000.

Analysis

If the relationship was created in *MakeView*, *Analysis* will identify variables without definition.

MakeView

Grid tables and relate buttons are examples of automatic relationships. *Analysis* does not load related tables automatically; therefore, the user must establish the relationship from a program or from the Relate dialog.

	Menu	Check	Analysis
REPEAT			
Property (A checkbox in the field dialog)		X	

Description When REPEAT is enabled, the variable will be initialized with the value taken from the previous record.

Comments This feature can save effort when entering items like STATE that remain constant with only a few exceptions over many records. Only the exceptions need be entered in detail.

	Menu	Check	Analysis
ROUTEOUT			
Command	X	X	X

Description Directs output to the named file until the process is terminated by CLOSEOUT. Output from commands such as FREQ and LIST is appended to the same output file as it is produced.

Syntax **ROUTEOUT** <filename>[**REPLACE**|**APPEND**]

Syntax Element	Description
<Filename>	An HTM document where the output is to be stored. If no directory is specified, the directory of the current project will be used.
REPLACE APPEND	If REPLACE is specified, any existing file of the same name is deleted prior to writing. If APPEND is specified or neither word is used, new output is appended to any existing file of the same name.

Comments Epi Info 2000 will send the output to an HTM document that can be read by any browser.

If no output is selected, Epi Info 2000 will create a new file typically

called OUTXXX.HTM where XXX is a sequential number. These files will be placed in the same directory as the current project.

The prefix for output files can be changes from the STORAGE options located in the *Analysis* command under Output.

Examples ROUTEOUT 'D:\EPI2000\Montly_Report.htm' REPLACE
ROUTEOUT 'D:\EPI2000\MyPage.htm' APPEND

See Also The storage facilities for creating a library of output files are described at the end of this chapter.

Menu Check Analysis

RUNPGM

Command X

Description Control is transferred to the second program, returning to the first automatically, beginning with the line following the RUN statement. Thus a program being RUN from another program is somewhat like an INCLUDE file or subroutine in other systems.

Syntax **RUNPGM** '<FileName>' : "<Program>"
RUNPGM '<FileName>'
RUNPGM "<Program>"

Syntax Element	Description
<FileName>	Path and filename for the MDB file or DOS .PGM file where the program is stored. If the path or filename contains a space, it must be enclosed in single quotes. If the program to be run is in the current project, the path need not be supplied.
<Program>	Program name. If the program name contains a space, it must be enclosed in double quotes. Not used for text files.

Comments Since the filename can include any path and database name, the program to be executed can be stored in a different database than the one in use.

Examples RUNPGM 'C:\EPI2000\SAMPLE.Mdb':"Test1"

Menu Check Analysis

SCREENTEXT

X

Command

Description

This places the specified text on the screen at the location specified, with the size, color, font, and style designated. Horizontal and vertical positions are given in percentage of screen width and height. Thus 50, 50 would be centered on the screen. If the item is in the middle 50% of screen width, it is centered. If Horizontal is less than 25, the text is left justified. When Horizontal is more than 75, the text is right justified. Sizes vary with different fonts, but can be determined through examining the font possibilities in *Eped*. Text Fonts must be enclosed in quotation marks and must be spelled exactly as they appear in a Windows font selection window, as in *Eped*.

Syntax

SCREENTEXT "<ItemText>", <Horizontal>, <Vertical>, <Size>, <Color>, <Style>

Syntax Element	Description
<Horizontal>	A number from 1 to 99 representing the distance from the left border of the window in percentage points.
<Vertical>	A number from 1 to 99 representing the distance from the top border of the window in percentage points.
<Size>	A number between 1 and 999 representing the font size
<Color>	Black Red Green Yellow
	Any of the fonts registered in the operating system of the computer. If the font is not available, text will be displayed using the default Windows font.
<Style>	Bold Magenta Underlying Cyan Italic White Blue

Comments

Screen text will be sent to the language module so that the user can have text available in different languages. See Multilanguage applications for more information.

Examples

ScreenText "Configurable Menu. See EPI2000.MNU", 3, 0, 8, yellow, Arial, Bold

Program-Specific Features

Menu

Screen text is a separated part of the MNU program. It should be closed with the command END. If END is missing, results can be unpredictable.

Menu Check Analysis

SELECT

Command

XX

Description

SELECT allows you to specify an expression that must be true for a record to be processed. If the current selection is "age > 35", then only those records with age > 35 will be processed. SELECT used alone without an expression will cancel all previous SELECT statements.

Syntax

SELECT <Expression>

Syntax Element	Description
<Expression>	Any valid Epi Info 2000 expression.

Comments

SELECT expressions are cumulative so that the two expressions:

```
SELECT AGE > 35
SELECT SEX = "F"
```

are equivalent to

```
SELECT (AGE>35) AND (SEX = "F")
```

Examples

```
SELECT ((age > 35) and (sex="F")) or ((age > 40)
and (sex="M"))
```

Note the use of parentheses to show relationships.

```
SELECT NAME = "MAC"
```

will select only "MAC", "Mac", etc., an exact match except for case.

```
SELECT NAME LIKE "MAC*"
```

will select "MAC", "MACDONALD", etc., an inexact match, in which the first three letters of NAME are "MAC".

Menu Check Analysis

SET

Command

X

Description

Includes in *Analysis* programs the settings that are made from the SET dialog in *Analysis*, so that these settings will be used whenever the program runs. SET provides various options that affect the performance of *Analysis* or its output.

Syntax

SET <Parameter>=<Value>

Syntax Element	Description
<Parameter> = <Value>	As many elements of this form as desired may appear in a single SET statement

Comments

Parameter	Values	Response
(-)	"<Text>"	In Boolean variables, NO will be represented as <Text>
(.)	"<Text>"	Variables with missing values will be represented as <Text>
(+)	"<Text>"	In Boolean variables, YES will be represented as <Text>
YN	"<Text1>", "<Text2>", "<Text3>"	Sets displayed text for Yes, No and Missing to Text1, Text2 and Text3 respectively
PROCESS	NORMAL DELETED BOTH	Deleted records are not included Only deleted records are included Deleted records are included
DELETED [alternative to PROCESS]	YES or (+) NO or (-) ONLY	Deleted records are included Deleted records are not included Only deleted records are included
FREQGRAPH	(+) Or ON	Show frequency graph next to the table
HYPERLINKS	(-) or OFF (+) or ON (-) or OFF	Do not include freq graph. Hyperlinks are included in the output Hyperlinks excluded
MISSING	(+) or ON (-) or OFF	Include missing values for analysis Do not include missing values for analysis
IGNORE (alternative to MISSING)	(+) or ON	Do not include missing values for analysis
SELECT	(-) or ON (+) Or ON (-) or OFF	Include missing values for analysis Display selection criteria Do not display selection criteria
SHOWPROMPTS	(+) Or ON	Displays the full prompt for variable name

STATISTICS	(-) or OFF NONE MINIMAL INTERMEDIATE COMPLETE	Displays Variable name only No statistics are displayed
PERCENTS	(+) or ON (-) or OFF	Percentages are included in TABLES output Percentages are excluded

Examples

SET HYPERLINKS=(+) PROCESS=UNDELETED (+)="True"
 (-)="False" (.)="Unknown" PERCENTS=(+) MISSING=(-)
 SHOWPROMPTS=(+) SELECT=(+) FREQGRAPH=(+)
 STATISTICS=INTERMEDIATE

Program-Specific Features

In the command generator, selecting "Save All" will generate a SET command, which contains all current settings. Selecting "Save Only" or "OK" will generate a SET command, which contains only changed settings. To force a SET command to be generated for a current value of a setting, change its value and then change it back again.

Menu Check Analysis

SETBUTTONS

X

Command

Description

Turns the buttons on the menu on or off.

Syntax

SETBUTTONS <Option>

Syntax Element	Description
<Option>	ON/OFF to display/hide buttons

Comments

Use to set up a user option for making buttons visible or invisible. If the buttons are currently visible, they are made invisible and inactive, and vice versa.

Examples

Program-Specific Features

		Menu	Check	Analysis				
SETPICTURE								
Command		X	X	X				
Description	Allows the user to choose an image and set it as the image for the current menu.							
Syntax	SETPICTURE <Image>							
	<table border="1"> <thead> <tr> <th>Syntax Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><Image></td> <td>Any Image in a valid format (BMP,GIF,JPG)</td> </tr> </tbody> </table>				Syntax Element	Description	<Image>	Any Image in a valid format (BMP,GIF,JPG)
Syntax Element	Description							
<Image>	Any Image in a valid format (BMP,GIF,JPG)							
Comments	The name and location of the chosen image is stored in the EPIINFO.INI file so that the menu can display the same image next time it runs. SETPICTURE does not set up a picture; it merely allows the user to do so.							

		Menu	Check	Analysis						
SORT										
Command				X						
Description	SORT allows you to specify the sequence in which records will appear when using LIST, GRAPH and WRITE commands. If no variable names are specified after the SORT command, the current sort is cleared, and subsequent output of records will be in their order in the original data table. If one variable name is given, the records will be sorted using that variable as the “key.” If more than one variable is specified, the records will be put in order by the first variable, then within a group with the same value of variable 1, the ordering will be by variable 2, etc.									
Syntax	SORT <VarName> [DESCENDING]									
	<table border="1"> <thead> <tr> <th>Syntax Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><Varname></td> <td>Variable to be sorted by</td> </tr> <tr> <td>DESCENDING</td> <td>Indicates that the sort order is descending; if not specified, ascending order will be used.</td> </tr> </tbody> </table>				Syntax Element	Description	<Varname>	Variable to be sorted by	DESCENDING	Indicates that the sort order is descending; if not specified, ascending order will be used.
Syntax Element	Description									
<Varname>	Variable to be sorted by									
DESCENDING	Indicates that the sort order is descending; if not specified, ascending order will be used.									
Comments	The parameter DESCENDING must be placed next to the variable to be sorted in descending order. Should several variables be sorted in descending order, one “DESCENDING” should be included for each of									

them.

Sort expressions are cumulative so that the two expressions:

```
Sort AGE
Sort SEX DESCENDING
```

are equivalent to

```
Sort AGE SEX DESCENDING
```

Examples

```
Sort AGE
Sort AGE DESCENDING Ill DESCENDING SEX DESCENDING
```

Menu Check Analysis

SOUNDEX

Property (A checkbox in the field dialog)

X

Description

SOUNDEX is a method of coding names so that differences in spelling are minimized. "Smith" and "Smyth" are both converted to the same SOUNDEX code, so that subsequent searches will find both names.

Menu Check Analysis

SYSINFO

Command

X

Description

This command displays a remarkable amount of information about the Windows 95, 98, NT, or 2000 operating system and the hardware and software of the computer. It includes type of processor, memory usage, disk space, programs currently active, date and version of all system modules, and many other items useful for solving problems.

Syntax

SYSINFO

Comments

SysInfo calls a Windows utility to display the information.

Menu Check Analysis

TABLES

Command

X

Description

TABLES does a cross-tabulation of the specified variables and sends the table to the screen, printer, or other current output. Values of the first variable will appear on the left margin of the table, and those of the second variable will be across the top of the table. Normally cells contain counts of records matching the values in the corresponding marginal labels. If the WEIGHTVAR parameter is given, the cells represent sums exactly as in SUMTABLES in Epi Info for DOS. TABLES DISEASE COUNTY WEIGHTVAR=COUNT gives the same results as SUMTABLES COUNT DISEASE COUNTY in Epi Info for DOS.

Syntax

TABLES <Exposure> <Outcome> {**STRATAVAR=** <Variable(s)>}
 {**WEIGHTVAR=** <Variable>} {**OUTTABLE =** <Table>}
 {**COLUMN SIZE =** <Number>} {**NOWRAP**}

Syntax Element	Description
<Exposure>	Variable in the database to be considered the risk factor (or * for all variables)
<Outcome>	Variable in the database considered Disease of consequence (or * for all variables).
<Variable>	Variable in the database
<Table>	A valid table name to be used to store output
<Number>	A number from 0 to 99 representing the number of columns to be displayed before splitting a table (default=6). If zero, all specified columns will be displayed without splitting the table.
NOWRAP	If present, no line breaks are inserted into text fields, so that columns are as wide as the longest line in the field.

Comments

For every possible combination of values of the strata variables, a separate table (stratum) for variable 1 by variable 2 will be produced. TABLES BAKEDHAM ILL STRATAVAR=SEX will produce a table of BAKEDHAM by ILL for each value of sex--one for M and one for F. TABLES BAKEDHAM ILL STRATAVAR=SEX RACE will produce a separate table of BAKEDHAM by ILL for each combination of SEX and RACE--female/black, female/white, male/black, male/white, etc.

If "*" is given instead of a variable name, each variable in the dataset is substituted for "*" in turn. Thus, if you would like to analyze each variable by illness status, the command:

TABLES * ILL

will produce tables of SEX by ILL, AGE by ILL, etc.

It is important to consider carefully before using "*" or requesting multidimensional tables, since there may be more tables than you want in terms of time, paper, and other costs. Pressing the Ctrl-Break keys will allow you to exit from a lengthy procedure if you change your mind.

The SET command can be used to determine the format of the tables by including percentage values in the tables (SET PERCENTS = ON).

Examples

TABLES disease county
 TABLES * COUNTY
 TABLES VANILLA ILL AGE SEX
 TABLES DISEASE COUNTY /SUM = COUNT

Menu Check Analysis

TYPEOUT

Command X

Description The TYPEOUT command inserts text, either a string or the contents of a file, into the output. Typical uses might include comments or boilerplate.

Syntax TYPEOUT "Text" [(<FontProperty>)] [**TEXTFONT** <Color> <Size>]
 TYPEOUT <FileName> (in single quotes if it contains spaces)

Syntax Element	Description
<Text>	Type text to be displayed

<FontProperty>	Properties (Underline, Bold, Italic) will be separated by commas (,)		
<Color>	Aqua	Lime	Red
	Black	Maroon	Silver
	Blue	Navy	Teal
	Fuchsia	Olive	White
	Gray	Purple	Yellow
	Green		
	#xxxxxx, where x is a hexadecimal digit; the pairs of digits represent the amount of red, green and blue, respectively, on a scale of 0-255, included in the color.		

Comments

TYPEOUT with a text string is similar to TITLE except that TYPEOUT places the text once when the command is encountered, while HEADER will appear at the top of each segment of output until it is cleared.

If no text in quotation marks follows the TYPEOUT command, TYPEOUT sends the contents of the file specified to the current output. This output can be in text or HTML.

Examples

TYPEOUT 'My Fancy Logo.htm'
Copies the contents of the file to output.

TYPEOUT "CONFIDENTIAL" (BOLD, UNDERLINE)
Displays the word "Confidential" in bold and underlined.

		Menu	Check	Analysis
UNDEFINE				
Command		X	X	X
Description	UNDEFINE removes a defined variable and any assigned values from the system.			
Syntax	UNDEFINE <Variable>			
	Syntax Element	Description		
	<Variable>	A Defined Variable		
Comments	A variable that already exists in the database cannot be undefined. To remove it from the database use WRITE * EXCEPT			
Examples				

Program-Specific Features

Check code

Undefine can be used for any defined variable at any time.

Analysis

Permanent variables cannot be undefined from the GEN screen. To Undefine a permanent variable, type the instruction and run it from the text editor.

Menu

Undefine can be achieved by assigning an empty string "" to the variable.

Menu Check Analysis

WRITE

Command

X

Description

WRITE will send records to an output table or file in the format that you specify. You can specify what variables will be written, the order in which they will appear, and the type of file to be written. Allowable output types are:

Syntax

WRITE <METHOD> {<Output Type>} {<Project>:}Table {<VarName(s)>}

WRITE <METHOD> {<Output Type>} {<Project>:}Table *
EXCEPT {<VarName(s)> }

Syntax Element	Description
<METHOD>	REPLACE APPEND
<Output Type>	See table below
<Project>	Path and filename

Output types available in Epi2000

Database type	Specifier	Elements
Jet database	"Access 97" or "Epi2000"	<path>:<table>
dBase III	"dBASE III"	<Path>
dBase IV	"dBASE IV"	<Path>
DBase 5.0	"dBase 5.0"	<path>
Paradox 3.x	"Paradox 3.x"	<Path>
Paradox 4.x	"Paradox 4.x"	<Path>

FoxPro 2.0	"FoxPro 2.0"	<Path>
FoxPro 2.5	"FoxPro 2.5"	<Path>
FoxPro 2.6	"FoxPro 2.6"	<Path>
Excel 3.0	"Excel 3.0"	<Path>
Epi Info 6	"Epi6"	<Path>
Excel 4.0	"Excel 4.0"	<path>
Text (Delimited)	"Text"	<Path>

Comments

Records deleted in *Enter* or SELECTed in *Analysis* are handled as in other *Analysis* commands, and defined variables may be written, allowing you to create a new Epi Info file that makes the changes permanent. Global and permanent variables will not be written unless explicitly specified.

If you wish to write only selected variables, they may be listed. The word EXCEPT may be inserted to indicate all variables except those following EXCEPT.

Either APPEND or REPLACE must be specified to indicate that an existing file/table by the same name will be erased or records will be appended to the existing file/table if present. If some but not all of the fields being written match those in an existing file during an APPEND, the unmatched fields will be added to the output table.

Examples

```
WRITE APPEND "Epi 2000" 'D:\EPI2000\demo1.MDB':table1
*
```

```
WRITE APPEND "Epi 2000" 'D:\EPI2000\demo1.MDB':test1 *
EXCEPT BOHID BOH ALIENNUMBE
```

Notes on the Output Library in *Analysis*

By default, output files are stored in the directory of the current project with a name composed of a prefix and a sequence number. The prefix and sequence number can be changed in the storage screen. The output of a single statistical procedure is called a result. Results accumulate in a given output file until the data source changes by READ, RELATE, SORT, SELECT or SET PROCESS, or until a CLOSEOUT command is executed.

Users can specify the location of the output by using the ROUTEOUT command. If no directory is given, the file is placed in the directory of the current project. Results accumulate until a CLOSEOUT command is executed.

Each result is logged to a results index file named IResults.htm, located in the project directory. Thus, if several projects share the same directory, they share the same results

index. Output files generated with HYPERLINKS set on will contain a link to the results index.

Results can be selected for archiving using the storage screen. Results so selected are logged to an archive index file named IArchive.htm. The archive index may be located in any directory, and may include references to results taken from more than one results index. Archived results are marked as archived in the results index. A result may be archived in more than one archive index, but this is not recommended.

Results can be selected for deletion using the storage screen. Because an output file can contain more than one result, it is not actually deleted until all its results have been deleted from the index. Archived results cannot be deleted when viewing the results index, and therefore are protected from deletion within the *Analysis* storage system. When viewing an archive index, archived results can be deleted from within the *Analysis* storage system, subject to the following limitations: 1) If a result is archived to more than one archive index, it can only be deleted from the first archive to which it was assigned, and deletion from the first archive will be effective even if the result has been archived elsewhere. 2) If a result is stored in an output file located in a different directory than its results index, it cannot be deleted from its results index.

Neither output files nor index files are protected from deletion using the operating system.

Storage is an interactive command only; no code is generated and its functions cannot be programmed. The storage screen looks like this:

Result Storage

Output File Prefix: Output File Sequence:

Results Folder:

Archive Folder:

Flag output files exceeding these limits

Age in Days: Number of Results: File Size (KB):

The output file prefix and sequence are set by the top two text boxes. The next sequence assigned will be one greater than that shown. Existing files of the same name are not overwritten; the sequence is incremented until an unused name is found.

The limits on age, number of results, and total file size are set by the bottom three text boxes. These are used in the Flag Files operation.

Selecting OK will close the window and save the current values of the five text boxes just discussed in the INI file. Selecting Close will close the window without saving the values.

The View, Delete, and Browse operations can be done on either a results index or an archive index. Since the names of these files are fixed, the user need only choose the folder in which they reside. The user chooses the file to be used by selecting the corresponding option button. The Archive operation can be done only on a results file; its button is disabled when the archive folder is selected.

Selecting Browse will open a common dialog file open window. The user must choose a file in the appropriate directory. Although new folders can be created, they cannot be selected because they will have no files in them; however, the user can then type in the folder name.

Selecting one of the other operations will open a new window, which looks the same for each command except that the operations available on it are different. The window for Archive looks like this:



Selecting Archive will cause the selected results to be archived. Selecting Archive All will cause all the results to be archived.

If the Delete operation was selected, the available operations are Delete and Delete All. Selecting Delete will cause the selected results to be deleted. Selecting Delete All will cause all the results to be deleted. Prior to deletion, a confirmation dialog will appear. Deletion will only affect the index file, unless all references to a particular output file are deleted. In that case, the confirmation dialog will indicate which files would be deleted.

If the View operation was selected, the available operations are Display and Flag Files. Selecting Display will cause the selected result to be displayed in the browser. Selecting Flag Files will cause a new window to be opened, which looks like this:

File Name	Results	Can Delete	Date	Size (KB)
D:\analysis\OUT239.htm	1	Yes	Dec 20 1999	5.6
d:\analysis\OUT240.htm	2	Yes	Dec 21 1999	9.7
d:\analysis\OUT241.htm	2	Yes	Dec 21 1999	9.6
d:\analysis\OUT242.htm	2	Yes	Dec 21 1999	7.6
d:\analysis\OUT243.htm	1	Yes	Dec 21 1999	4.1
d:\analysis\OUT244.htm	1	Yes	Dec 21 1999	5.1
d:\analysis\OUT245.htm	1	Yes	Dec 21 1999	2.1
d:\analysis\OUT246.htm	7	Yes	Jan 3 2000	13.8
d:\analysis\OUT247.htm	1	Yes	Jan 3 2000	15.3
d:\analysis\OUT248.htm	1	Yes	Jan 3 2000	4.8
d:\analysis\OUT249.htm	1	Yes	Jan 3 2000	4.9
d:\analysis\OUT250.htm	1	Yes	Jan 3 2000	4.9
d:\analysis\OUT251.htm	1	Yes	Jan 3 2000	4.9
d:\analysis\OUT252.htm	1	Yes	Jan 3 2000	4.9

Buttons: Help, Delete, Flag On/Off, Close

On this screen, for each file, the following information is displayed: its name; the number of results it contains; whether it can be deleted (true if results in it are not archived elsewhere); file date; and file size in KB. The file date, number of results, and file size columns are highlighted if they exceed the limits set on the main storage screen. The number of results and file size limits are cumulative; the accumulation is done first for non-deletable files, then for deletable files; within these classes, more recent files are done first. If a file exceeds all three limits, its name is highlighted initially. Selecting Flag On/Off inverts the highlighting on deletable files. Selecting Delete deletes all highlighted files; the actual deletion is preceded by a confirmation dialog.

Functions and Operators



Modifying Values Within Expressions

OPERATORS

[ARITHMETIC](#)

[COMPARISON](#)

BOOLEANS

[AND](#)

[OR](#)

[XOR](#)

[NOT](#)

NUMERIC

[EXP](#)

[SIN, COS, TAN](#)

[LOG](#)

[LN](#)

[ABS](#)

[RND](#)

[TRUNC](#)

[ROUND](#)

[NUMTODATE](#)

[NUMTOTIME](#)

DATE FUNCTIONS

[YEARS](#)

[MONTHS](#)

[DAYS](#)

[YEAR](#)

[MONTH](#)

[DAY](#)

SYSTEM FUNCTIONS

[SYSTEMDATE](#)

[SYSTEMTIME](#)

[ENVIRON](#)

[EXISTS](#)

[FILEDATE](#)

TIME FUNCTIONS

[HOURS](#)

[MINUTES](#)

[SECONDS](#)

[HOUR](#)

[MINUTE](#)

[SECOND](#)

TEXT FUNCTIONS

[TXTTONUM](#)

[TXTTODATE](#)

[SUBSTRING](#)

[UPPERCASE](#)

[FINDTEXT](#)

[FORMAT](#)

Overview

Expressions are combinations of literal values, variables, functions, and operators that can be evaluated to a single result. Expressions can be used in *MakeView* (Check code) and *Analysis*. Within an expression, the values of variables can be modified by a number of functions and operators. .

An Expression consists of one or more Operands (variables or literal values) and one or more Operators (like +, -, *, and /). Expressions, no matter how complex, can eventually be evaluated to produce a single value, like 1.323, or “True” or “False.”

Functions modify the value of one or more variables to produce a result. For example, ROUND (2.33333) produces the value 2. Operators are used to combine two items. For example, the “+” operator combines Var1 and Var2 to produce a sum, as in Var3=Var1+Var2.

The many functions and operators available in Epi Info 2000 are described in this chapter. Almost all functions require arguments enclosed in parentheses and separated by commas. Where arguments are required, there may not be any spaces between the function name and the left parenthesis.

Operators

Arithmetic (+ - * / ^ MOD)

Operators

Description	(See individual descriptions below, under Syntax.)	
Syntax	[Expression] <Operator> [Expression]	
	Syntax Element	Description
	<Operator>	^ Exponentiation * , / Numeric Multiplication or Division MOD Modulus or Remainder + , - Numeric addition or subtraction & String Concatenation
	[Expression]	A numeric value or a variable containing data in numeric format
Comments	The result will be expressed in numeric format. The basic mathematical operators that can be used in Epi Info 2000 are Addition (+), Subtraction (-), Multiplication (*), Division (/), Exponentiation (^), and Modulo or Remainder (MOD). Mathematical operators can be used in combination with other commands. Arithmetic operators are shown in descending order of precedence. Parentheses can be used to control the order in which operators are evaluated, although frequently the default order will achieve the right result.	
Example	ASSIGN <Var1> = 1250 MOD 100 Variable <Var1> will have the value of 50 (the remainder after 1250 is divided by 100).	
See Also	Comparison Operators, Boolean Operators	

COMPARISON (> >= = <> < <= LIKE)

Operators

Description	(See individual descriptions below, under Syntax.)	
Syntax	[Expression] <Operator> [Expression]	
	Syntax Element	Description
	<Operator>	= Equal to > Greater than < Less than >= Greater than or equal to <= Less than or equal to <> Not equal to

	LIKE	Left side variable matches right side pattern; in pattern, * matches any number of characters, ? matches any one character
Comments	[Expression]	Any valid expression
	<p>Comparison operators can be used in If Then and Select statements in both Check code and the <i>Analysis</i> program. Yes/No variables can only be tested for equality against the yes/no constants (+), (-) and (.). Comparison operators will be executed from left to right. There is no hierarchy among them.</p> <p>The <> operator can be used only with numeric variables. For non numeric variables use NOT</p>	
Example	<pre>READ 'D:\EPI2000\SAMPLE.MDB':VIEW OSWEGO SELECT AGE < 15</pre> <p>Records selected will include ages 0 to 14 years.</p>	
See Also	NOT, SELECT	

Boolean Operators (AND, OR, XOR, NOT)

The Boolean operators can be used only in logical expressions used in IF-THEN or SELECT commands. A logical expression is the result of a comparison operation or a Boolean combination of logical expressions.

AND

Operator

Description	Logical AND. If both conditions are true, it returns true; otherwise, it returns false.	
Syntax	[Logical Expression] AND [Logical Expression]	
	Syntax Element	Description
	[Expression]	Any valid logical expression in Epi Info 2000
Comments		
Example	<pre>READ FILE1 DEFINE RESULT IF AGE < 18 AND SEX = "F" THEN RESULT = "GIRL" END</pre>	

Age	Sex	Result
5	F	GIRL
7	M	<Missing>
18	F	<Missing>
21	F	<Missing>
27	M	<Missing>

In this case the value of "GIRL" will be assigned to all records that meet both criteria (Age < 18) and (sex = "F").

See Also OR, XOR, NOT

OR

Operator

Description	Logical OR. Returns true if either of the expressions is true.	
Syntax	[Logical Expression] OR [Logical Expression]	
	Syntax Element	Description
	[Expression]	Any valid logical expression in Epi Info 2000

Comments

Example

```

READ C:\EPI2000\SAMPLES.MDB:VIEWOSWEGO
DEFINE ICE_CREAM
IF VANILLA = (+) OR CHOCOLATE=(+) THEN
    ICE_CREAM = (+)
ELSE
    ICE_CREAM = (-)
END
    
```

VANILLA	CHOCOLATE	ICE_CREAM
Yes	Yes	Yes
No	Yes	Yes
Yes	No	Yes
No	No	No
Yes	Yes	Yes

See Also AND, XOR, NOT, IF

XOR (eXclusive OR)

Operator

Description	Logical XOR. Returns true if one and only one value is true, and false if both are true or both are false	
Syntax	[Logical Expression] XOR [Logical Expression]	
	Syntax Element	Description
	[Expression]	Any Valid logical Expression in Epi Info 2000

Comments

Example

```

READ C:\EPI2000\SAMPLE.MDB:VIEWOSWEGO
DEFINE ONE_ICE_CREAM
IF VANILLA = (+) XOR CHOCOLATE = (+) THEN
    ONE_ICE_CREAM = (+)
ELSE
    ONE_ICE_CREAM = (-)
END
    
```

VANILLA	CHOCOLATE	ONE_ICE_CREAM
Yes	Yes	No
No	Yes	Yes
Yes	No	Yes
No	No	No
Yes	Yes	No

See Also

AND, OR, NOT, IF

NOT

Operator

Description	Reverses the True or False value of the logical expression that follows	
Syntax	NOT [Expression]	
	[Expression]	Description
		Any valid logical expression in Epi Info 2000
Comments	If the value of an expression is TRUE, NOT returns the value FALSE. If the expression is FALSE, NOT <expression> is TRUE.	

Example

```

READ C:\EPI2000\SAMPLE.MDB:VIEWOSWEGO
DEFINE NO_VANILLA
IF NOT VANILLA = (+) THEN
    NO_VANILLA = (+)
ELSE
    NO_VANILLA = (-)
END
    
```

VANILLA	NO_VANILLA
Yes	No
No	Yes

See Also

AND, OR, NOT, IF

Numeric Functions

EXP

Function

Description	Raises the base of the natural logarithm (e) to the power specified	
Syntax	EXP(<Variable>)	
		Description
Comments	<Variable>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant
Example	<pre>READ 'D:\EPI2000\SAMPLE.MDB': VIEWOSWEGO DEFINE EXP_A EXP_A = EXP(AGE) LIST EXP_A</pre>	
See Also	LOG, LN	

SIN, COS, TAN

Function

Description	Returns the respective trigonometric value for the specified variable	
Syntax	SIN(<Variable>)	
		Description
	<Variable>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant
Comments	The variable will be interpreted as the angle in radians.	
Example	<pre>READ 'D:\EPI2000\SAMPLE.MDB': VIEW OSWEGO DEFINE SIN_A SIN_A = SIN(AGE) LIST SIN_A</pre>	
See Also	To convert degrees to radians, multiply by pi (3.1415926535897932) divided by 180	

LOG

Function

Description	Returns the base 10 logarithm (decimal logarithm) of a numeric value or variable. If the value is zero or null, it will return a null value.	
Syntax	LOG(<Variable>)	
	Syntax Element	Description
	<Variable>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant
Comments	The result will be numeric.	
Example		

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEWOSWEGO
DEFINE DEC_LOG
DEC_LOG = LOG (AGE)
LIST AGE DEC_LOG
```

See Also LN

LN

Function

Description Function LN will return the natural logarithm (logarithm in base e) of a numeric value or variable. If the value is zero or null, it will return a null value.

Syntax LN(*<Variable>*)

Syntax Element	Description
<i><Variable></i>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant

Comments The result will be numeric.

Example

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEWOSWEGO
DEFINE NATLOGOFAGE
NATLOGOFAGE = LN(AGE)
LIST AGE NATLOGOFAGE
```

See Also LOG

ABS

Function

Description Function ABS will return the absolute value of a variable by removing the negative sign, if any.

Syntax ABS(*<Variable>*)

Syntax Element	Description
<i><Variable></i>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Comments The result will be numeric.

Value	ABS Function
-2	2
1	1
0	0
-0.0025	0.0025

Example

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEWOSWEGO
DEFINE AGE2
DEFINE AGE3
AGE2 = AGE * -1
AGE3 = ABS (AGE2)
```

LIST AGE AGE2 AGE3

See Also TRUNC

RND

Function

Description RND will generate a random number between *<Var1>* and *<Var2>*.
Syntax **RND(** *<Min>* , *<Max>* , [*<Seed>*])
 Syntax Element Description

Syntax Element	Description
<i><Min></i>	A number or numeric variable that corresponds to the lowest value for the random number to be generated
<i><Max></i>	A number or numeric variable that corresponds to the highest possible value for the random number to be generated
<i><Seed></i>	Seed for random sequence. Giving the same seed causes a series of random numbers to be the same, that is, to start at the same point. Each successive number is then random, but repeating the listing will generate the same series. This feature is not implemented in the current version.

Comments The result will be a decimal number. It can be converted to integers with the TRUNC function.

Example

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEWOSWEGO
DEFINE Random1
DEFINE Random2
DEFINE Random3
Random1=RND(1,100,1)
Random2=RND( 1,100,2)
Random3=RND(1,100,3)
LIST Random1 Random2 Random3
```

See Also TRUNC

TRUNC

Function

Description Removes decimals from a numeric variable, returning the integer part of the number. Can be thought of as “rounding toward zero.”
Syntax **TRUNC**(*<Variable>*)

	Syntax Element	Description
	<i><Variable></i>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.
Comments	The result will be returned in numeric format.	
Example	<pre> READ 'D:\EPI2000\SAMPLE.MDB': VIEWOSWEGO DEFINE Random1 DEFINE Random2 DEFINE Random3 Random1=RND(1,100,1) Random2=RND(1,100,2) Random3=RND(1,100,3) DEFINE TRC1 DEFINE TRC2 DEFINE TRC3 TRC1 = TRUNC(Random1) TRC2 = TRUNC(random2) TRC3 = TRUNC(RANDOM3) LIST TRC1 RANDOM1 TRC2 RANDOM2 TRC3 RANDOM3 </pre>	
See Also	ROUND	

ROUND

Function

Description Rounds the number stored in the variable to the closest integer. Positive numbers are rounded up to the next higher integer if the fractional part is greater than or equal to 0.5. Negative numbers are rounded down to the next lower integer if the fractional part is greater than or equal to 0.5

Syntax **ROUND** (*<Variable>*)

	Syntax Element	Description
	<i><Variable></i>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.
Comments	The result will be returned in numeric format.	

Differences between ROUND and TRUNC

Value	Trunc	Round
0.123456	0	0
7.999999999	7	8
45.455	45	46

Example

```

READ 'D:\EPI2000\Sample.Mdb':viewOswego
FREQ AGE
DEFINE Decade
Decade = ROUND(AGE/10) + 1
                    
```

See Also LIST AGE Decade
TRUNC

NUMTODATE

Function

Description Transforms three numbers into a date format.

Syntax NUMTODATE(<Year>,<Month>,<Date>)

Syntax Element	Description
<Year>	A numeric variable or a number representing the year
<Month>	A numeric variable or a number representing the month
<Day>	A numeric variable or a number representing the day

Comments If the date resulting from the conversion is not valid (e.g., December 41, 2000), the date will be recalculated to the corresponding valid value (e.g., January 10, 2001). When <Year> ranges between 0 and 29, it will be represented as the respective year between 2000 and 2029, while values from 30 to 99 will be represented as the respective year between 1930 and 1999. The oldest date that can be recorded is Jan 01, 100.

Day	Month	Year	Date created
02	02	1999	02/02/1999
60	01	1999	03/01/1999
15	18	2000	03/18/2001
99	99	99	06/07/0107
20	08	74	08/20/1974

Example

```

READ 'D:\epi2000\My File.mdb':viewMain
DEFINE Day1
DEFINE month1
DEFINE year1
day1= day(dob)
month1 = month(dob)
year1 = year(dob)
define date2
date2= NUMTODATE(year1,month1,day1)
    
```

See Also FORMAT

NUMTOTIME

Function

Description Transforms three numbers into a time or date/time format.

Syntax NUMTOTIME (<Hour>, <Minute>, <Second>)

Syntax Element	Description
<Hour>	Numeric constant or variable representing hours

<Minute> Numeric constant or variable representing minutes
<Second> Numeric constant or variable representing seconds

Comments Time must be entered in 24-hour format. Invalid dates will be recalculated to the respective valid time. If the number of the hour exceeds 24, the resulting variable will have a date/time format and the default day 1 will be December 31, 1899.

Hour	Minute	Second	Time created
00	00	00	12:00:00 AM
00	00	90	12:01:30 AM
15	84	126	04:26:06 PM
25	00	00	12/31/1899 1:00:00 AM
150	250	305	01/05/1900 10:15:05 AM
15000	7500	8954	09/21/1901 07:29:14 AM

Example

```
READ 'D:\EPI2000\SAMPLE.MDB':VIEWOSWEGO
DEFINE VAR3
VAR3 = SYSTEMTIME
DEFINE HOUR1
DEFINE MINUTE1
DEFINE SECOND1
HOUR1 = HOUR(VAR3)
MINUTE1 = MINUTE(VAR3)
SECOND1 = SECOND(VAR3)
DEFINE TIME2
TIME2 = NUMTotime(HOUR1,MINUTE1,SECOND1)
LIST VAR3 TIME2
```

See Also NUMTODATE, FORMAT

Date and Time Functions

YEARS

Function

Description Returns the number of years from *<VAR1>* to *<VAR2>* in numeric format. If any of the variables or values included in the formula is not a date, the result will be Null.

Syntax YEARS(*<VAR1>*, *<VAR2>*)

Syntax Element	Description
<i><Var1></i>	Variable in date format
<i><Var2></i>	Variable in date format

Comments If the date stored in Var1 is more recent than that in Var2, the result will be the difference in years expressed as a negative number.

Example

```
READ 'C:\EPI2000\SAMPLE.mdb':viewSURVEILLANCE
DEFINE TODAY
TODAY = '01/01/2000'
DEFINE AGE_YEARS
AGE_YEARS = YEARS (DOB, TODAY)
MEANS AGE_YEARS
```

See Also YEAR, MONTH, MONTHS, DAY, DAYS

MONTHS

Function

Description Returns the number of months between *<Var2>* and *<Var1>*. If any of the variables or values included in the formula is not a date, the result will be Null.

Syntax MONTHS(*<VAR1>*, *<VAR2>*)

Syntax Element	Description
----------------	-------------

<i><Var1></i>	Variable in date format
---------------------	-------------------------

<i><Var2></i>	Variable in date format
---------------------	-------------------------

Comments If the date stored in Var1 is older than that in Var2, the result will be the difference in months expressed as a negative number.

Example

```

READ 'C:\EPI2000\MYFILE.MDB':VIEW DATES
DEFINE AGE_MONTHS
AGE_MONTHS = MONTHS (DOB, 01/01/2000)
LIST AGE_MONTHS
    
```

See Also YEAR, YEARS, MONTH, DAY, DAYS

DAYS

Function

Description Returns the number of days between *<Var2>* and *<Var1>*. If any of the variables or values included in the formula is not a date, the result will be Null.

Syntax DAYS(*<VAR1>*, *<VAR2>*)

Syntax Element	Description
----------------	-------------

<i><Var1></i>	Variable in date format
---------------------	-------------------------

<i><Var2></i>	Variable in date format
---------------------	-------------------------

Comments If the date stored in Var1 is more recent than that in Var2, the result will be the difference in days expressed as a negative number.

Example

```

READ 'C:\EPI2000\MYFILE.MDB':VIEW DATES
DEFINE AGE_DAYS
AGE_DAYS = DAYS (DOB, 01/01/2000)
LIST AGE_DAYS
    
```

See Also YEAR, YEARS, MONTH, MONTHS, DAY

YEAR

Function

Description Extracts the year from a date.

Syntax YEAR(*<Variable>*)

Syntax Element	Description
----------------	-------------

<i><Variable></i>	Variable in date format
-------------------------	-------------------------

Comments If the date is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```

READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE CURR_YEAR
CURR_YEAR = YEAR(01/01/1990)
LIST CURR_YEAR
    
```

See Also YEARS, MONTH, MONTHS, DAY, DAYS

MONTH

Function

Description Extracts the month from a date.

Syntax **MONTH**(<Variable>)

Syntax Element	Description
<Variable>	Variable in date format

Comments If the date is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```

READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE CURR_MONTH
CURR_MONTH = MONTH(01/01/1990)
LIST CURR_MONTH
    
```

See Also YEAR, YEARS, MONTHS, DAY, DAYS

DAY

Function

Description Extracts the day from a date.

Syntax **DAY**(<Variable>)

Syntax Element	Description
<Variable>	Variable in date format

Comments If the date is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```

READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE Curr Day
Curr_Day = DAY(01/01/1990)
LIST CURR_Day
    
```

See Also YEAR, YEARS, MONTH, MONTHS, DAYS

Time Functions

HOURS

Function

Description Returns the number of hours between *<VAR2>* and *<VAR1>* in numeric format.**Syntax** **HOURS**(*<VAR1>*, *<VAR2>*)

Syntax Element

Description

<Var1>

Variable in time or date/time format

<Var2>

Variable in time or date/time format

Comments

If the date stored in *<Var1>* is older than that in *<Var2>*, the result will be the difference in hours expressed as a negative number. Both variables must contain data in date, time, or date/time format. If any of the variables or values included in the formula is not a date, the result will be Null.

Example

```
READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE HOUR1
HOUR1=HOURS(DATE,DTOFARR)
LIST HOUR1
```

See Also

HOUR, MINUTES, MINUTE, SECONDS, SECOND

MINUTES

Function

Description Returns the number of minutes between *<VAR2>* and *<VAR1>* in numeric format.**Syntax** **MINUTES**(*<VAR1>*, *<VAR2>*)

Syntax Element

Description

<Var1>

Variable in time or date/time format

<Var2>

Variable in time or date/time format

Comments

If the date stored in *<Var1>* is older than that in *<Var2>*, the result will be the difference in minutes expressed as a negative number. Both variables must contain data in date, time, or date/time format. If any of the variables or values included in the formula is not a date, the result will be Null.

Example

```
READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE MIN1
MIN1=MINUTES(DATE,DTOFARR)
LIST MIN1
```

See Also

HOURS, HOUR, MINUTE, SECONDS, SECOND

SECONDS

Function

Description Returns the number of seconds between *<VAR2>* and *<VAR1>* in numeric format.

Syntax	SECONDS(<VAR1>, <VAR2>)						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Syntax Element</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"><i><Var1></i></td> <td style="padding: 2px;">Variable in time or date/time format</td> </tr> <tr> <td style="padding: 2px;"><i><Var2></i></td> <td style="padding: 2px;">Variable in time or date/time format</td> </tr> </tbody> </table>	Syntax Element	Description	<i><Var1></i>	Variable in time or date/time format	<i><Var2></i>	Variable in time or date/time format
Syntax Element	Description						
<i><Var1></i>	Variable in time or date/time format						
<i><Var2></i>	Variable in time or date/time format						
Comments	If the date stored in <i><Var1></i> is older than that in <i><Var2></i> , the result will be the difference in seconds expressed as a negative number. Both variables must contain data in date, time or date/time format. If any of the variables or values included in the formula is not a date, the result will be Null.						
Example	<pre> READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO DEFINE SEC1 SEC1=SECONDS(DATE,DTOFARR) LIST SEC1 </pre>						
See Also	HOURS, HOUR, MINUTES, MINUTE, SECOND						

HOUR

Function

Description	Returns a numeric value that corresponds to the hour recorded in a date/time or time variable.				
Syntax	HOUR(<Variable>)				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Syntax Element</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"><i><Variable></i></td> <td style="padding: 2px;">Variable in date format</td> </tr> </tbody> </table>	Syntax Element	Description	<i><Variable></i>	Variable in date format
Syntax Element	Description				
<i><Variable></i>	Variable in date format				
Comments	If the time is stored in a Text variable, the function will not be processed and the result will be Null.				

Example

```

READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE Local
Local = SYSTEMTIME
LIST Local
DEFINE hour1
hour1=hour(local)
LIST Local hour1
LIST Local
LIST hour1
    
```

See Also HOURS, MINUTES, MINUTE, SECONDS, SECOND

MINUTE

Function

Description	Returns a numeric value that corresponds to the minute recorded in a date/time or time variable.				
Syntax	MINUTE (<i><Variable></i>)				
	<table border="1"> <thead> <tr> <th>Syntax Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i><Variable></i></td> <td>Variable in date/time or time format</td> </tr> </tbody> </table>	Syntax Element	Description	<i><Variable></i>	Variable in date/time or time format
Syntax Element	Description				
<i><Variable></i>	Variable in date/time or time format				
Comments	If the time is stored in a Text variable, the function will not be processed and the result will be Null.				

Example

```

READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE Local
Local = SYSTEMTIME
LIST Local
DEFINE min1
min1= minute(local)
LIST Local min1
LIST Local
    
```

See Also HOURS, HOUR, MINUTES, SECONDS, SECOND

SECOND

Function

Description Returns a numeric value that corresponds to seconds recorded in a date/time or time variable.

Syntax SECOND(<Variable>)

Syntax Element

Description

<Variable>

Variable in date/time or time format

Comments If the time is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```
READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE Local
Local = systemtime
LIST Local
DEFINE sec1
sec1= second(local)
LIST Local sec1
```

See Also HOURS, HOUR, MINUTES, MINUTE, SECONDS

Text Functions

TXTTONUM

Function

Description Returns a numeric value that corresponds to the string.

Syntax TXTTONUM(<Variable>)

Syntax Element

Description

<Variable>

Variable in text format

Comments

Example

Incorrect:

```
VAR1 = TXTTONUM(VAR1)
```

Correct:

```
DEFINE VAR2
VAR2 = TXTTONUM (VAR1)
```

See Also TXTTODATE, FORMAT, SUBSTRING

TXTTODATE

Function

Description Returns a numeric value that corresponds to the string.

Syntax TXTTODATE(<Variable>)

Syntax Element

Description

	<i><Variable></i> Variable in text format
Comments	The text variable can be in any format that can be recognized as a date, e.g. "Jan 1, 2000", "1/1/2000".
Example	Incorrect: VAR1 = TXTTODATE (VAR1) Correct: DEFINE VAR2 VAR2 = TXTTODATE (VAR1)
See Also	TXTTONUM, FORMAT, SUBSTRING

SUBSTRING

Function									
Description	SUBSTRING(<i><Variable></i> , [First], [Length])								
Syntax	<table border="1"> <thead> <tr> <th>Syntax Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i><Variable></i></td> <td>Variable in text format</td> </tr> <tr> <td>[First]</td> <td>The position of the first character to extract from the file</td> </tr> <tr> <td>[Length]</td> <td>The number of characters to extract</td> </tr> </tbody> </table>	Syntax Element	Description	<i><Variable></i>	Variable in text format	[First]	The position of the first character to extract from the file	[Length]	The number of characters to extract
Syntax Element	Description								
<i><Variable></i>	Variable in text format								
[First]	The position of the first character to extract from the file								
[Length]	The number of characters to extract								
Comments	SUBSTRING cannot be used with non-string variables.								
Example	<pre> READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO DEFINE TEXT1 TEXT1 = "01/01/1999" DEFINE YEAR1 YEAR1 = TXTTONUM(SUBSTRING(TEXT1,7,4)) MEANS YEAR1 </pre>								
See Also	TXTTONUM, FORMAT, TXTTODATE								

UPPERCASE

Function					
Description	Transforms the content of a string (text) variable to uppercase letters.				
Syntax	<table border="1"> <thead> <tr> <th>Syntax Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i><Variable></i></td> <td>Variable in text format</td> </tr> </tbody> </table>	Syntax Element	Description	<i><Variable></i>	Variable in text format
Syntax Element	Description				
<i><Variable></i>	Variable in text format				
Comments					
Example	<pre> READ 'D:\EPI2000\MyFile.Mdb':viewDemographics DEFINE LASTNAME2 LASTNAME2 = UPPERCASE(LASTNAME) LIST LASTNAME2 LASTNAME </pre>				
See Also	SUBSTRING, FINDTEXT				

FINDTEXT

Function

Description	Returns the position in a variable in which the string is located.						
Syntax	FINDTEXT (<Variable1>,<Variable2>)						
	<table border="1"> <thead> <tr> <th>Syntax Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><Variable1></td> <td>String of characters to be found</td> </tr> <tr> <td><Variable2></td> <td>String to be searched in</td> </tr> </tbody> </table>	Syntax Element	Description	<Variable1>	String of characters to be found	<Variable2>	String to be searched in
Syntax Element	Description						
<Variable1>	String of characters to be found						
<Variable2>	String to be searched in						
Comments	If the string is not found, the result is 0; otherwise it is a number corresponding to the position of the string starting from the left.						
Example	<pre> READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO DEFINE VAR11 VAR11=FINDTEXT("H",LASTNAME) LIST LASTNAME VAR11 </pre>						
See Also	SUBSTRING, TRUNC, FORMAT						

FORMAT

Function

Description	Changes the format of one variable type to another one						
Syntax	FORMAT (<Variable> [, "Format specification"])						
	<table border="1"> <thead> <tr> <th>Syntax Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><Variable></td> <td>Variable in any format</td> </tr> <tr> <td>[Format Specification]</td> <td>Any of the following words <ul style="list-style-type: none"> <i>a) Date formats</i> <ul style="list-style-type: none"> General Date 11/11/1999 05:34 Long Date System's Long date format Medium Date System's Medium date format Short Date System's Short date format Long Time System's Long time Medium Time System's Medium time Short time System's Short time <i>b) Number formats</i> <ul style="list-style-type: none"> General Number No thousand separator Currency Thousand separator plus two decimal places (based on system settings) Fixed Standard At least #.### Percent #,###.### Scientific Number multiplied by 100 plus a percent sign Scientific Standard scientific notation </td> </tr> </tbody> </table>	Syntax Element	Description	<Variable>	Variable in any format	[Format Specification]	Any of the following words <ul style="list-style-type: none"> <i>a) Date formats</i> <ul style="list-style-type: none"> General Date 11/11/1999 05:34 Long Date System's Long date format Medium Date System's Medium date format Short Date System's Short date format Long Time System's Long time Medium Time System's Medium time Short time System's Short time <i>b) Number formats</i> <ul style="list-style-type: none"> General Number No thousand separator Currency Thousand separator plus two decimal places (based on system settings) Fixed Standard At least #.### Percent #,###.### Scientific Number multiplied by 100 plus a percent sign Scientific Standard scientific notation
Syntax Element	Description						
<Variable>	Variable in any format						
[Format Specification]	Any of the following words <ul style="list-style-type: none"> <i>a) Date formats</i> <ul style="list-style-type: none"> General Date 11/11/1999 05:34 Long Date System's Long date format Medium Date System's Medium date format Short Date System's Short date format Long Time System's Long time Medium Time System's Medium time Short time System's Short time <i>b) Number formats</i> <ul style="list-style-type: none"> General Number No thousand separator Currency Thousand separator plus two decimal places (based on system settings) Fixed Standard At least #.### Percent #,###.### Scientific Number multiplied by 100 plus a percent sign Scientific Standard scientific notation 						

Yes/No	Displays NO if number = 0, else displays Yes
True/False	False if number = 0 True if number <> 0
On/Off	Displays 0 if number = 0, else Displays 1
Custom format	Allows user to create customized formats

Comments

Output may vary based on specific configuration settings of the computer.
Format(Time, "Long Time")

```
MyStr = Format(Date, "Long Date")
MyStr = Format(MyTime, "h:m:s") ' Returns "17:4:23".
MyStr = Format(MyTime, "hh:mm:ss AMPM") ' Returns "05:04:23 PM".
MyStr = Format(MyDate, "dddd, mmm d yyyy") ' Returns "Wednesday, ' Jan 27 1993".
```

' If format is not supplied, a string is returned.
MyStr = Format(23) ' Returns "23".

' User-defined formats.
MyStr = Format(5459.4, "##,##0.00") ' Returns "5,459.40".
MyStr = Format(334.9, "###0.00") ' Returns "334.90".
MyStr = Format(5, "0.00%") ' Returns "500.00%".
MyStr = Format("HELLO", "<") ' Returns "hello".
MyStr = Format("This is it", ">") ' Returns "THIS IS IT".
MyStr = Format("This is it", ">:*") ' Returns "THIS IS IT".

For more information about format, please refer to Microsoft Access or Microsoft Visual Basic documentation.

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
DEFINE date2
DEFINE date3
DEFINE date4
DEFINE date5
DEFINE date6
DEFINE date7
DEFINE date8
DEFINE date9
DEFINE date10
DEFINE date11
date2=format(BOH, "Currency")
date3=format(BOH, "fixed")
date4=format(BOH, "Standard")
date5=format(BOH, "Percent")
date6=format(BOH, "Scientific")
date7=format(BOH, "Yes/No")
date8=format(BOH, "True/false")
date9=format(BOH, "On/Off")
date10=format(BOH, "VB\s #,###.##")
list dob date2 date3 date4 date5 date6 date7 date8 date9
date10
```

See Also TXTONUM, TXTTODATE, NUMTODATE,

System Functions

SYSTEMDATE

Function

Description Returns the date stored in the computer's clock.
Syntax SYSTEMDATE
Comments SYSTEMDATE cannot be changed (assigned) from *Analysis*. To use the systemdate for computations, a new variable must be defined. For example, to calculate next week's date:

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
DEFINE TODAY_DATE
TODAY_DATE = SYSTEMDATE + 7
```

See Also SYSTEMTIME

SYSTEMTIME

Function

Description Returns the time stored in the CPU clock at the time the command is executed.
Syntax SYSTEMTIME
Comments SYSTEMTIME cannot be changed from *Analysis* (assign). To use the system time for computations, a new variable must be defined. For example, to calculate a time 2 hours after current time:

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
DEFINE LATER
LATER = SYSTEMTIME + (720 / 24 / 60 / 60)
```

See Also Variable LATER stores current time plus 720 seconds (2 hours).
SYSTEMDATE

ENVIRON

Function

Description Returns the value of a DOS Environment variable, such as PATH or COMSPEC.
Syntax ENVIRON (<Name of DOS Environment Variable>)

Syntax Element	Description
----------------	-------------

<Variable>	Variable in text format
------------	-------------------------

Comments

Example

```
IF NOT EXISTS(ENVIRON("TEMP") &"\BATCH.TXT" THEN
  DIALOG "Input file does not exist" TITLETEXT="Error"
  QUIT
END
```

See Also

EXISTS

Function

Description	Returns TRUE if a file exists; otherwise returns FALSE.
Syntax	EXISTS(<i><Variable></i>)
	Syntax Element
	Description
	<i><Variable></i> Complete file path and name in text format
Comments	May return FALSE if user does not have permission to access the file.
Example	IF EXISTS ("C:\Epi2000\Test.BAT") then Execute 'C:\epi2000\test.bat' End
See Also	EXECUTE

FILEDATE

Function

Description	Returns the date a file was last modified (or created).
Syntax	FILEDATE (<variable>)
	Syntax Element
	Description
	<i><Variable></i> Complete file path and name in text format
Comments	This function might be useful when several users are updating a large database.
Example	READ 'C:\epi2000\Hepatitis.mdb'viewHepatitis DEFINEdfine New_Update global ASSIGN New_Update = filedate('c:\epi2000\update1.mdb') If filedate('c:\epi2000\Hepatitis.mdb') > New_Update then Dialog "The information might be out of date. Please check your source" titletext="Warning" end

See Also

Summary of Functions

Syntax	Description
<Var1> = <Var2> + <Var3>	<Var1> Stores the sum of <Var2> and <Var3>
<Var1> = <Var2> - <Var3>	<Var1> Stores the difference between <Var2> and <Var3>
<Var1> = <Var2> * <Var3>	<Var1> Stores the product of <Var2> times <Var3>
<Var1> = <Var2> / <Var3>	<Var1> Stores the difference of <Var2> times <Var3>
<Var1> = <Var2> ^ <Var3>	<Var1> Stores the result of <Var2> to the power of <Var3>
<Var1> = <Var2> MOD <Var3>	Returns The remainder for <var2> divided by <var3>
<Var1> = <Var2> AND <Var3> = <Var4>	True if both conditions are met
<Var1> = <Var2> OR <Var3> = <Var4>	True if any of those conditions are met
<Var1> = <Var2> XOR <Var3> = <Var4>	
NOT <Var1> = <Var2>	True if <Var2> is not equal to Var1
<Var1> = <Var2>	True when <Var1> has the same value or structure than <Var2>
<Var1> <> <Var2>	True when <Var1> has different value or structure than <Var2>
<Var1> < <Var2>	True when <Var1> is smaller than <Var2>
<Var1> <= <Var2>	True when <Var1> is smaller or equal <Var2>
<Var1> > <Var2>	True when <Var1> is greater than <Var2>
<Var1> >= <Var2>	True when <Var1> is greater equal than <Var2>
TXTTONUM (<Variable>)	
TXTTODATE (<Variable>)	
FORMAT (<Variable>,"[Type]")	"general" Produces string variables
SUBSTRING (<Variable>,First, Length)	First is the first character to include. Length is the number of characters
UPPERCASE (<Variable>)	
FINDTEXT ([String],<Variable>) [Operator] position	
YEARS (<Date1> , <Date2>)	Return the difference between <Date2> and <Date1> in years
MONTHS (<Date1> , <Date2>)	Return the difference between <Date2> and <Date1> in Months
DAYS (<Date1> , <Date2>)	Return the difference between <Date2> and <Date1> in Days
HOURS (<Date1> , <Date2>)	Return the difference between <Date2> and <Date1> in Hours
YEAR (<Variable>)	Returns the Year (in Numeric format) from a date or date/time variable.
MONTH (<Variable>)	Returns the Month (in Numeric format) from a date or date/time variable.
DAY (<Variable>)	Returns the Day (in Numeric format) from a date or date/time variable.
HOURL (<Variable>)	Returns the Hour (in Numeric format) from a date or date/time variable.
MINUTE (<Variable>)	Returns the Minute (in Numeric format) from a date or date/time variable.
SECOND (<Variable>)	Returns the Second (in Numeric format) from a date or date/time variable.
MDY (<Variable>,<Variable> , <Variable>)	Transforms three numbers into a date string
<Variable> = SYSTEMDATE	Assigns the Date to <Variable>
<Variable> = SYSTEMTIME	Assigns current time to <Variable>
HMS (<Variable>,<Variable> , <Variable>)	
RND (<Var1> , <Var2> , <Var3>) .	<Var1> represents min output, <Var2> max output, and <Var3> optional seed).
Log (<Variable>)	Calculates the logarithm in base 10
Ln (<Variable>)	Calculates the logarithm of Variable in base n
EXP (<Variable>)	
TRUNC (<Variable>)	Removes decimals from <Variable>
ROUND (<Variable>)	Rounds the value of <Variable>
ABS (<Variable>)	Removes sign from negative values
<Function> (<Variable>)	Returns the trigonometric function in radians
RECNUMBER ()	
RECDELETED ()	
ENVIRON (EnvVar)	Returns the value of the environment variable EnvVar. Both the argument and the result are strings.
EXISTS (File)	Returns true if the file File exists. File is a string and Exists is a YesNo
FILEDATE (File)	Returns the modification date of the file File, or null if File does not exist. File is a string and FileDate is date.
<Variable> = (.)	(.) Represents missing values

Internal Details



For Visual Basic Programmers

Overview

This chapter provides details on selected parts of the Epi Info internal architecture for Visual Basic programmers who wish to understand how Epi Info 2000 Views are stored or how to write new statistical modules for incorporation into future versions of Epi Info. Views provide metadata about the databases created by Epi Info and also describe the screen appearance of forms sufficiently so that they can be used, for example, to construct an HTML equivalent of an Epi Info View. The Broad Street Library of Statistics is the name given to an anticipated library of statistical routines built around the specification included in this chapter for the IEPI statistical interface. IEPI provides standard means for sending data and settings to a statistical module (a Windows Dynamic Linked Library – DLL, or other ActiveX module) and receiving back the statistical results, both in the form of an HTML string suitable for display and alternatively as a series of named parameters (e.g., "p value") and values (e.g., 0.005).

The Structure of Views

A View is a Microsoft Access file with a structure defined by Epi Info. It contains information on the screen appearance of a database or part of a database. In this sense, it resembles the questionnaire file and especially the header of the REC file in Epi Info for DOS.

Each field in a View is described by a record in the View table. User-generated Check code and settings for the field are stored in the same record. View records generally contain:

- The Number of the Page where the field is displayed. Page 0 is reserved for special records that pertain to the entire View.
- The Prompt for display on the screen.
- The horizontal and vertical locations of the Prompt and the Field itself, as percentages of screen height and width.
- The Size of the field as displayed (although text fields scroll to accept 128 characters regardless of their size on the screen, and multiline fields will accept almost unlimited amounts of text).
- Check code for the field.
- Lists of miscellaneous items, such as code files.
- The location where data items are stored or retrieved. The DataTable field contains a reference to a record on Page 0 that contains the actual data table name.

The name of the field is assumed to be the same as the field name stored in the View.

Records with 0 or negative page numbers contain additional information, such as colors and the names of background images that apply to whole pages or to the entire View.

Special field records contain information on Related Views and Grids, which are a type of related View.

Each Data source (Database and DataTable) used by the View has a field in Page 0 of Type SOURCE and a Name consisting of DATA plus a number (1, 2, or 3, for example). The numbers start at 1 and proceed in the order of first use of the data source by the View. The data source fields of the SOURCE records contain the path, filename, and TableName of the data table, using paths relative to the location of the .MDB that contains the View.

Data source fields in each Field record contain the Name used by the SOURCE record, for example, DATA1.

A similar method is used for Code Tables, storing the location of the Code Table in records of Type SOURCE, with Names CODE1, CODE2, etc. Field records refer to CODE1, CODE2, etc., where the actual names are stored.

Related Views are identified in records on Page 0 with Type SOURCE and names RELVIEW1, RELVIEW2, etc. The same names are used to identify the related views in Field Records of type RELATE.

The effect of these indirect references is to concentrate information on data sources (metadata) in records of type SOURCE. To use a different SOURCE, all that is required is to change the SOURCE record so that it refers to another source. If fields in the source referred to match those in the fields of the View, then the link is successful; otherwise, the fields can be ignored or an error message issued.

Version information for a View should be included on Page 0 of the view (e.g., "Epi Info 2000, Version 1.00").

This process provides a method of substituting new data sources for those in a View. A MetaData table containing the names of SOURCE records in a View will contain references to an alternative set of data sources and the name of the View. MetaData files can be loaded directly into either *Enter* or *Analysis* to bring up the relevant View and Data tables. Source records in the View are essentially ignored in favor of the information in the MetaData file (or table). The MetaData information can also be used to generate a description of the database for other purposes, although related Views must be consulted to find information on a complex hierarchy.

Records in the MetaData table are identical to the SOURCE records in a VIEW, with the addition of one more record of Type "VIEW" that identifies the main View to be used with the designated SOURCE files or tables.

The user can OPEN a MetaData table in *Enter* or *MakeView*, and READ a MetaData table in *Analysis*. In *MakeView*, the MetaData table provides only the View. In *Enter* and *Analysis*, the SOURCE records are read from the MetaData table and the information is stored in memory before the designated VIEW is read. SOURCE records in the VIEW are

ignored if there is a MetaData table, thus effectively substituting the data sources specified in the MetaData table for those specified in the View table. This allows a single view to be used with as many data sources as desired, merely by making a new MetaData table for each View/Source combination. Views and Sources must “match” to the extent that variables referred to in the View are present in the Source and related tables have appropriate keys.

A Simplified View Table

PageNo	Name	Type	DataBase	DataTable	DataField
0	DATA1	SOURCE	SAMPLE.MDB	recOswego	
0	DATA2	SOURCE	SAMPLE.MDB	recOswego1	
0	CODE1	SOURCE	..\Codes\CODES.MDB	NYCounties	
0	RELVIEW1	SOURCE	SAMPLE.MDB	recContacts	

A Simplified MetaData Table for the View OSWEGO in SAMPLE.MDB

PageNo	Name	Type	DataBase	DataTable	DataField
0	MAINVIEW1	VIEW	SAMPLE.MDB	viewOSWEGO	
0	DATA1	SOURCE	SAMPLE.MDB	recDecatur	
0	DATA2	SOURCE	SAMPLE.MDB	recDecatur1	
0	CODE1	SOURCE	..\Codes\CODES.MDB	GACounties	
0	RELVIEW1	SOURCE	SAMPLE.MDB	recGAContacts	

NOTE: References to a data source outside of the current MDB must be replaced by a reference to an Access Link table within the current MDB.

Fields in a View Table	
PageNo	Number
Prompt	May be very long
Name	Name of field—up to 12 characters
Type	Field type—up to 12 characters *
Index	Type of index if any—Unique, secondary, etc.
DSize	Size of displayed field on screen
FSize	Maximum size of field data
Format String	String of specified types
PlocX	Location of prompt LEFT
PlocY	Location of prompt TOP
FlocX	Location of field LEFT
FlocY	Location of field TOP
TabOrder	Tab order For discussion—page, entire form, or one form
PFont	Prompt Font
PFontsize	Prompt Font Size
PFonttype	Prompt Font Type (italic, etc)
FFont	Field Font
FFontsize	Field Font Size
FFonttype	Field Font Type (italic, etc)
Lists	Legal values, code lists, and special purpose info for some field types, (Very large capacity)
Checkcode	Epi Info Commands for this field
Database	Link to project database (default) or to another database
Datatable	Link to data table
Datafield	Link to datafield

A Sample View Containing All Field Types Available in Epi Info 2000

The records in the view have been sorted by PageNumber and distributed over several pages to fit this document. The notes at the bottom of the columns and in fine print are not part of the database.

PageNumber	Name	Prompt	Type	Index
-2			Background	0
-1			Background	0
0	GRIDVIEW1		SOURCE	
0	RELVIEW1		SOURCE	
0	DATA1		SOURCE	
0	GENERAL		INFO (To be inserted)	
1	ThisIsText	This is Text	TEXTBOX	
1	ThisIsALabel	This is a Label	TEXTONLY	
1	ThisIsTextuppercase	This is Text(uppercase)	UPPERCASE	
1	ThisIsAMultiline	This is a Multiline	MULTILINE	
1	ThisIsANumber	This is a Number	NUMBER	

	1 ThisIsPhone	This is Phone Number	PHONENUMBER	
	1 ThisIsDate	This is Date	DATE	
	1 ThisIsTime	This is Time	TIME	
	2 ThisIsDateTime	This is date time	DATETIME	
	2 ThisIsCheckbox	This is checkbox	CHECKBOX	
	2 ThisIsYesno	This is YesNo	YES/NO	
	2 ThisIsOption	This is Option	OPTFRAME	
	2 ThisIsAuto	This is Auto ID	AUTOIDNUM	
	2 ThisIsACommand	This is a command button	COMMANDBUTTON	
	2 ThisIsImage	This is Image	IMAGE	
	2 ThisIsGrid	This is Grid	GRID	
	2 ThisIsRelate	This is Relate	RELATE	
Page 0 contains records referring to the entire view. Positive page numbers refer to actual pages. Negative pages contain items that pertain to an entire page, such as a background image.	Field Name. This must be unique within the View and cannot contain spaces.	The prompt contains text that precedes the field. It can contain spaces.	The field type. Note that GRID and RELATE fields refer to related Views and indirectly to related data tables.	Not currently used for data fields. Stores color of background for pages.

Dsize	Fsize	Formatstring	Plocx	Plocy	Flocx	Flocy
1.609658	3.350515	0 0=image in corner 1=whole page image	7545 To be removed	6480 To be removed		
1.609658	3.350515	0	7545 To be removed	6480 To be removed		
		Epi Info 2000, Version 0.95	7545 Height in Twips of original page	6480 Height in Twips of original page		
0.152247	0.048969		5.003353	5	19.08786	5
0	0		40	0	0	0
0.193159	0.048969		5.030181	14.94845	30.38229	14.94845
0.241449	0.090206		5.030181	25	24.34608	25
0.096579	0.048969	###;###	5.003353	40	24.11804	40
0.193159	0.048969	###-###-####;###- ###-####	5.030181	50	29.17505	50
0.160966	0.048969	##-##-####;MM-DD- YYYY	5.030181	60.05154	19.5171	60.05154
0.209256	0.048969	##-##-## ??:HH:MM:SS AMPM	5.003353	70	19.49027	70
0.193159	0.048969	##-##-####-##-##-## ??:MM-DD-YYYY HH:MM:SS AMPM	5.030181	10.05155	23.54125	10.05155
			8.423876	20	5.003353	20
0.112676	0.048969		5.030181	29.89691	21.32797	29.89691
0.18994	0.245704	R 1 3 4 R=Option text on Right 1=XXXXXX 3=Number of options 4=Array Index of frame	5.030181	39.94845	5.003353	43.35051
0.096579	0.048969	;	5.030181	75	21.73038	75
0.279946	0.064433		5.003353	85	5.003353	85
0.162978	0.126289		45.07042	20.10309	45.07042	26.66667
0.420523	0.208763	GRIDVIEW1	40.04024	45.10309	40.04024	50.44674
0.225352	0.064433		25.75453	14.43299	54.99664	85
Size of Prompt label in % of page width	Width of data entry field		Upper left corner of Prompt in % of page width	Top of Prompt in % of page height	Upper left corner of Entry Field in % of page width	Top of Entry Field in % of page height

Taborder	Pfont	Pfontsize	Pfonttype	Ffont	Ffontsize	Ffonttype
	1 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	0 MS Sans Serif	8	0 False			
	3 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	4 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	5 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	6 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	7 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	8 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	1 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	2 MS Sans Serif	8	0 False			
	3 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	4 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	5 MS Sans Serif	8	0 False	MS Sans Serif	8	0 False
	7 MS Sans Serif	8	0 False	Arial		
	8 MS Sans Serif	8	0 False	MS Sans Serif		
	9 MS Sans Serif	8	0 False	recgridTest1 ThisIsGrid		
	38 MS Sans Serif	8	RELVIEW1	MS Sans Serif		
Order in which cursor visits fields	Font for Prompt	Font size for Prompt	Font type for Prompt-- Values for BOLD and ITALIC. Here neither is set, giving normal type.	Font for Entry Field	Font size for Entry Field	Font "type" for Entry Field

Lists	Checkcode	Database	Datafield	Datatable
		Testall.MDB Testall.MDB Testall.MDB		recTest1ThisIsGrid viewThisIsRelate recTest1 DATA1
				DATA1 DATA1 DATA1 DATA1 DATA1 DATA1 DATA1 DATA1
<p>Yellow;Pink;Green ; Option captions</p> <p>One;Two;Three; Fields in Grid TO BE REMOVED. Many Allow 1 or Many records to be entered in related view. Lists are kept here for special purposes.</p>	<p>Contains the text of all Check code stored by field. Special records (not shown here) are generated to store Check code for BeforeRecord, After Record, etc. All variable definitions are stored in ????</p>	<p>Testall.MDB</p> <p>Name of MDB in which data table resides</p>		<p>DATA1 DATA1 DATA1 DATA1</p> <p>DATA1 DATA1 DATA1 DATA1 -- SHOULD BE GRIDVIEW1</p> <p>RELVIEW1</p> <p>Unique name generated by MakeView to indicate the Name of the record on Page 0 containing the actual reference to a data table.</p>

The Broad Street Library of Statistics: Specifications for Statistical, Graphing, and Mapping Modules for Epi Info 2000

The statistics, graphing, and mapping in Epi Info 2000 are implemented as ActiveX (formerly OLE) modules. The interface specification used is available for others to use in developing additional modules that could be added to a future version of Epi Info 2000. Communication between a calling module and an ActiveX module such as TABLE.DLL is done through an Interface, which the ActiveX module IMPLEMENTS. The standard Interface Class for Epi Info 2000 is called IEPI (Letter “I” EPI, not numeral “1” EPI), found in the Visual Basic program file called IEPI.CLS and the Interface Module, IEPI.DLL.

If the ActiveX module is written in Visual Basic, Version 6, the module includes the words, IMPLEMENTS IEPI, in its declarations. Selecting each of the IEPI methods or properties in the programming module of Visual Basic 6 will copy the skeleton of the property or method into the new module, with the addition of “IEPI_” prior to the name of the property or function. Thus GetFunctionNames becomes IEPI_GetFunctionNames, etc. The IEPI.DLL file should be copied to the programmer’s computer and registered for use in the system registry, either through Visual Basic or by using a separate program like REGSVR32.EXE.

The IEPI Type Library, Source Code for a DLL Containing Virtual Functions to be IMPLEMENTED by Compatible ActiveX Modules

```
Public Property Get FunctionNames() As Variant
'Returns the names of functions in the module as an array of strings
End Property
```

```
Public Property Let NumRows(ByVal vNewValue As Long)
'Set to the number of rows in the dataset by the client that provides the data
End Property
```

```
Public Property Let NumColumns(ByVal vNewValue As Long)
'Set to the number of columns in the dataset by the client that provides the data
End Property
```

```
Public Property Let NumStrata(ByVal vNewValue As Long)
'Set to the number of strata in the dataset by the client that provides the data
End Property
```

```
Public Property Let ColumnTypes(ByVal vNewValue As Variant)
'An array set to the type of data contained in each column
End Property
```

```
Public Property Let ColumnNames(ByVal vNewValue As Variant)
'An array set to the name of each column
End Property
```

```
Public Property Let Settings(ByVal vNewValue As Variant)
'Settings relevant to the procedures being requested
'Is an array with two dimensions, (1..2,1..n). Settings[1,n] is the name of the setting;
'Settings[2,n] is its value. Both are variants.
End Property
```

```
Public Property Let DataArray(ByVal vNewValue As Variant)
'An array having the dimensions (1..columns,1..rows, 1..strata) containing
'the data. If the data are not stratified, the third dimension and the value of the index are
'always 1. If only a single value, such as a Key, is passed, then columns, rows, and strata
'are all 1, and DataArray[1,1,1) = the Key.
End Property
```

```
Public Property Let MoreDataComing(ByVal vNewValue As Boolean)
'A boolean set by the client if more data will follow that is not currently
'in the DataArray.
End Property
```

```
Public Property Get Explanation() As String
'Additional explanation, if any, for display
End Property
```

```
Public Property Get ResultArray() As Variant
'Results expressed as name of statistic, value of statistic, in a
'two-dimensional array or collection. May be used as an alternative to the
'string returned by DoFunction
End Property
```

```
Public Function DoFunction(FunctionName As String) As String
'Calls on the stated routine. Note that routines must be
'Public functions so that they remain in memory while
'needed; this routine allows them to be called
'by passing the string name of the routine. Returns
```

'Resultstring if successful; otherwise, an error code.
End Function

Public Property Get Settings() as Variant

'Settings relevant to the procedures being requested. This should be implemented as a 2-dimensional array (1..2, 1..n) with the first item being a string "setting name" and the second the default value of the setting. This allows the client program to discover or display the setting possibilities even without documentation of the server module.
End Property

Interaction of the Client (e.g., *Analysis* or *StatCalc*) with the Server (e.g., STAT01.DLL, containing Fisher Exact and Odds Ratio Functions) proceeds as follows:

- Client checks the FunctionNames property of the STATXX module (say, Fisher Exact, for example) to get a list of available methods.
- *Analysis* then constructs an SQL query that includes the SELECTS, IF's, and other conditions prescribed by the user, and places the data in the DataArray property, specifying the number of Rows, Columns, and Strata in the corresponding properties. Setting, or the name of a settings file, can be passed in the SETTINGS property.
- The client program, *Analysis* in this case, then calls the statistical function.
- The Statistical Server (User-supplied function) checks for essential parameters (data, for example) and if some are missing, pops up a user dialog to get those needed.
- The Statistical Server (User-supplied function) does appropriate calculations and returns results as:

A string (up to 32 K), with results suitable for display on the screen or for printing. This can be in plain text format, but the preferred format is HTML, so that the result displays nicely in a browser. Strings suitable for translation into non-English languages should be enclosed between tags with the name "tlt" (for "TransLaTion"), as follows:
<tlt>Confidence Limits</tlt>. These tags are not displayed by common browsers, but can be used by Epi Info programs *Analysis* and *StatCalc* to identify items requiring translation. Each item is sent to the language translation module before being sent to the HTML output file in the *Analysis* or *StatCalc* program.

An array containing:

Name of Statistic	Value
etc.	
for the statistics produced.	

Analysis or *StatCalc* formats and displays the results that are returned as a string or array. If the module has its own display, as for Chart, Map, and Summary, the user interacts with the server module until he/she wishes to return to the client.

Common Features of IEPI Modules

These are ActiveX modules that are callable from *Analysis* or *StatCalc*, and use the IEPI interface to obtain data from the calling program

Each module has a number of properties by which it is configured, that can be set from dialogs (property pages) within the module and then saved. The name of the saved properties can be passed in the SETTINGS property of IEPI when the module is called. Note that this resembles the mechanism by which .MAP files store the properties of *Epi Map*, allowing one to recreate the map merely by referring to the name of the .MAP file.

The name of the data file or an SQL string to read it may be passed to the module, or a complete set of instructions may be stored in a .MAP or .CHT file for use by the module.

Limitations and Problem Resolution



How Far Toward the Rainbow Can We Go?

Limits

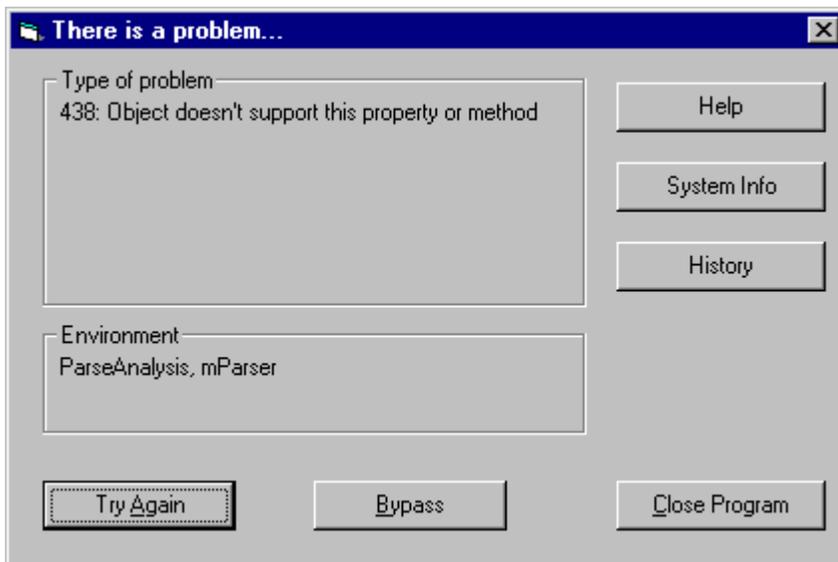
Epi Info 2000 inherits some limitations from the Windows environment and from microcomputers in general. A fairly fast processor (100 MHz or faster) is required for pleasant operation. For slower processors in particular, increasing Random Access Memory (RAM) beyond 32 or 64 megabytes will speed up program operation, although, at current prices, a new and faster computer may be a better choice.

Specific limitations in Epi Info 2000 include:

Number of fields per Questionnaire or View	A Microsoft Access table is limited to 256 fields, but Epi Info automatically creates additional related Views to remove this limitation.
Number of records in a data table	Practically limited only by disk space.
Number of tables in a .MDB database file	1000
Number of characters in a text field	128
Number of characters in a multiline field	Theoretically 2 gigabytes, but you may get tired of typing or run out of disk space first

Problem Resolution

Since many “errors” are not the fault of the user, and can result from circumstances outside the current program, we prefer to call them “problems,” and seek to resolve them as quickly as possible. Most of the programs in Epi Info contain a problem resolution module for handling “errors.” When a problem arises that cannot be resolved by the program automatically, a dialog such as the following may appear.



The number and description of a Microsoft Visual Basic Error is displayed. At this point, several options are available. You can choose to BYPASS the error and see if the program functions normally. If the problem is a corrupt data file, for example, this may allow you to continue work. Sometimes a number of problem messages will occur one after the other, but clicking on BYPASS for each will eventually allow you to return to safe ground. If not, then choosing CLOSE PROGRAM will terminate the current program. You may lose work that has not been saved, but this may be necessary if BYPASS does not work. Since *Enter* and *MakeView* save each page automatically, the loss should not be great.

If a problem occurs repeatedly, further information can be obtained from the Problem module by choosing HISTORY or SYSTEM INFORMATION. HISTORY gives a list of the internal program functions that were called prior to the occurrence of the problem. The list may be helpful to the Epi Info programming team in resolving a problem. SYSTEM INFORMATION brings up a Windows function that provides a huge amount of information about the Windows operating system as it exists on the current computer at the time of the problem. SYSTEM INFORMATION can be used to determine the name, version, and date of all the Dynamic Linked Libraries (DLLs) on the computer, for example. SYSTEM INFORMATION is also available under SETTINGS on the main Epi Info 2000 menu.

If all else fails and a Windows program is hopelessly locked up, the time-honored “Ctrl-Alt-Delete” key combination should bring up a Windows dialog that shows what programs are running. It may indicate that one is “not responding,” and allow the program to be terminated. It may also indicate that there are several instances of a single program in memory, and allow you to terminate the extra ones. This can be useful in resolving mysterious problems in writing to a file, caused by the fact that it is already being accessed by another copy of the same program.

Windows and DOS files have an attribute that can be set to “Read-Only.” This protects the file from changes, but also prevents its use as a database and prevents replacing it by copying another file of the same name into the same folder. To see the attributes of a file,

use My Computer on the Windows desktop to find the file, right-click on the icon or name, and choose “Properties.” A check box in the property window indicates whether the file is “Read-Only.” Clicking in the check box will change the status to “unchecked” or writable. A grayed-out check mark means that some, but not all, of the files selected before the right click had read-only attributes set. In My Computer, Ctrl-A can be used to select all files within the current folder; you can then right-click and choose Properties to examine the status of all files in the folder.

Before contacting the Epi Info Helpline, it is reasonable to reboot Windows to see if a problem is resolved. It is also a good idea to ascertain that other programs are operating properly on the same computer, and that the Local Area Network to which you may be connected is not having temporary problems.

If these steps do not resolve the problem, contact information for the Epi Info Helpline is provided on the title page of this manual. Careful notes on the problem and on the computer environment in which Epi Info is installed will help in problem resolution. When reporting problems, please include the date of the Epi Info distribution version found on the bottom of the main menu.

The INI File

Epi 2000 creates a file called EPIINFO.INI. Epi Info 2000 programs use the INI file to store configuration settings. For the end user, the INI file is almost invisible. The INI file is organized in sections identified by square brackets []. There is a section for each Epi Info 2000 program, and for each MNU file created and run in the Epi2000.EXE menu. Permanent variables are stored in a section called VARIABLES.

If all programs included with Epi2000 are not installed initially and the you decide to add them later with the installation program, the program will detect the existent INI file and it will prompt whether to replace it or not. If the INI file is replaced, permanent variables will be lost and may have to be reset within application programs.

In general, if the INI file is lost or deleted, every program will re-create its own entries when it is executed next time; however, in some cases an error can occur. For example, if the INI file is deleted or renamed the Menu.exe will ask for the Image before displaying the menu window. Once an image is selected, the menu will store it in the re-created INI file. When the wrong syntax is used in an entry the program may not recognize it and it will assume that there is no entry for that property. Although we do not recommend editing the INI file, it may be possible to remove a problem by doing so, particularly if a corrupt entry can be found for a file path. Generally removing a single-line entry from the INI file will not do harm except to make a program use a default value for the item in question. It is very important when editing an INI file to be sure it is saved in plain DOS text format—not in rich text, WordPerfect, or Microsoft Word formats. The Wordpad editor seems to detect and save plain text format satisfactorily.

A sample INI file looks like this, although the contents of the one on your computer may be quite different:

```
[MnuEPI2000]
Buttons=True
Picture=D:\EPI2000\Epi2000.JPG
TopLoc=0
LeftLoc=0
Height=90
Width=90
[VARIABLES]
CompDate=3/3/2000
Language=U.S. English
[Analysis]
Most Recent Database=Sample.mdb
Process Records=UNDELETED
[Storage]
[EpiInfo 2000]
Represent Yes as=Yes
Represent No as=No
Represent Missing as=Missing
[Statistics]
Statistics Type=INTERMEDIATE
[Output]
Show Complete Prompt=1
Show Selection=1
Show Graphics=1
Show Percents=1
Include Missing=0
[Nutstat]
Age=Yes
AgeUnits=Yrs&Mos
Years=
TodaysDate=Yes
BirthDate=Yes
BirthDateTag=mm/dd/yyyy
Table=nutchildren
ImportTable=
ZScores=Yes
Percentiles=Yes
PercOfMed=No
RepTemplate=nutReport.vrd
[Translations]
Lang1=English
```

Translation



Notes for Translators

Translating the Epi Info 2000 Programs

Epi Info 2000 is designed for easy translation into non-English languages. All the English phrases that can safely be translated are contained in tables in the database LANGUAGE.MDB. Commands and other reserved words in the *Analysis* and *Check* programming languages should not be translated, although the screen prompts that lead to their generation can be translated. Each Epi Info 2000 program has its own table in a LANGUAGE.MDB database (e.g., the phrases from *Analysis* are found in the *Analysis* table). Error messages common to several programs are found in the ERROR table. Each *Menu* (.MNU) file used in the menu produces a separate table in the database, which can be translated. English versions of these menu tables are generated automatically when a user creates a new MNU file.

Each language has its own directory of the installed Epi Info 2000 directory. Every installation has at least the directories ..\ENGLISH\ and ..\ENGLISH\HELP\, containing the English versions of the program phrases (c:\Epi2000\ENGLISH\LANGUAGE.MDB, for example) and the English versions of the help files and manual. Exercises such as OSWEGO may also be located in subdirectories of the ..\ENGLISH\ directory. An installation program called TSETUP.EXE is provided to create proper directories for a new language and install the translated files.

To add a new language, first examine the LANGUAGE.MDB database, using *VisData* or Microsoft Access, to appreciate how the database is constructed. To make a translation in SPANISH, for example, first create a subdirectory called ..\SPANISH\ under the Epi Info installed directory (usually ..\EPI2000\). Make another subdirectory under ..\SPANISH\ called HELP. Then copy the English version of LANGUAGE.MDB to ..\SPANISH\. Use *MakeView* to create a new field called SPANISH in each of the program tables in LANGUAGE.MDB. Your translation should include the tables for each program and the ERRORS table. When you or a user create a new MNU file for the Epi Info 2000 *Menu* program, a new table is automatically created with a column containing the phrases from the MNU so that a translation can be provided in LANGUAGE.MDB. Now that you have a View and a Data table for each program, translating the program into Spanish is simply a matter of placing the Spanish translations beside the English phrases in the same record in the database. If no translation is needed, leave the item blank, and the program will automatically default to English. This would be appropriate for single characters or an “=” sign, for example.

After each translation is complete, set the LANGUAGE variable in EPIINFO.INI to the name of the desired column (e.g., Spanish) and then run the program. This can be done either by editing EPIINFO.INI or by editing the EPI2000.MNU file to allow the user to set the LANGUAGE variable equal to SPANISH. Several examples of these settings are provided in EPI2000.MNU.

After the LANGUAGE variable has been set to SPANISH, the programs should show Spanish phrases automatically rather than the English equivalents. Check the spacing of phrases on the programs screen and make appropriate changes if crowding occurs. A beta test should be conducted with representative users, to test the function of the programs and the clarity and correctness of the translation.

After the translation is complete, the Epi Info Team will insert it into the database for the next version of Epi Info if you send it to the Epi Info Help Line at epiinfo@cdc.gov.

Translating the Manual and Help Files

The manual and help files were written in Microsoft Word 2000 and saved as HTML files. They can be reconverted to Microsoft Word format by Word 2000, translated, and then saved once again as HTML files. If they are placed in the proper directory (e.g., ..\SPANISH\HELP\), they will be automatically displayed when the LANGUAGE variable is set to SPANISH through the choice on the main menu, which in turn sets LANGUAGE to SPANISH in the EPIINFO.INI file.

LANGUAGE.MDB in the Main Epi Info 2000 Directory

To demonstrate language capability, a single LANGUAGE.MDB is supplied with Epi Info 2000 that contains pseudotranslations in Spanish, French, Italian, Portuguese, and German. These translations were done by an inexpensive computer program (Easy Translator 2) and are not intended for serious use. The Epi Info language module looks first for LANGUAGE.MDB in the directory for the currently chosen language (e.g., SPANISH), and, if one is not found, looks in the main Epi2000 directory. Hence, if you provide a real translation for even a few of the programs, the machine translation will not be used.

Advantages of Having Separate Directories for Each Language

If you are preparing a translation for further distribution, DO NOT PLACE THE TRANSLATION IN THE MULTICOLUMN LANGUAGE.MDB in the main directory. Although doing so provides a quick demonstration mechanism, the system described above, with separate directories for each language, has many advantages and allows translators in various parts of the world to distribute their language modules independently, perhaps from a web site, without having to send them to Atlanta for incorporation into the core Epi Info system. Revisions can be made for any language without disturbing other translations.

Preparing Files for Installation by TSETUP.EXE

The Language database and help files must be placed in the proper directories and compressed (“zipped”) into a single, self-expanding module (a self-expanding EXE file)

that can be run by the TSETUP.EXE program supplied with Epi Info 2000. The exact nature of the self-expanding EXE files is not crucial, but they must create directories of the correct name under the directory from which they are run. A program called ARJ is provided that may be used for compression.

The compressed executable file should be given the name of the language it contains—SPANISH.EXE in our example—and the file should be placed in the same directory with TSETUP.EXE. A translation of TSETUP.EXE itself can be provided by translators for end users, if desired.

In the following paragraphs, the language Spanish is used for example. However, any language name of eight characters or less is acceptable, and may be substituted for “Spanish.” In file names, case is not important, so .EXE is equivalent to .exe, and may be used interchangeably.

Translators

Initially, place language database (language.mdb), HELP files, and any other necessary files for a particular language in a source directory or subdirectory. For clarity, give the source directory the language name. For example, place all files and subdirectories required for the Spanish translation into the subdirectory c:\source\Spanish.

Then, compress (“zip”) all files and subdirectories residing in this source directory into a single, self-expanding module, called a “self-expanding,” or “self-extracting,” .EXE file. Use a public domain compression program such as ARJ (supplied with Epi Info 2000, with license) to create the self-expanding .EXE file – one .EXE for each language.

Use the compression program to assign the compressed executable file (.EXE) the name of the language, consistent with the name of the source subdirectory from which the non-compressed files are copied: for example, Spanish.exe.

Using the compression program, place the resulting .EXE file in a subdirectory, named TRANSEXEXE, of the directory which contains TSETUP. For each language the user wishes to employ, TSETUP will expect a file with that language name, suffixed by .EXE, in the TRANSEXEXE subdirectory, within the directory in which TSETUP resides.

Users

With TSETUP, a component program of Epi Info 2000, the user can then initiate the self-expanding .EXE file for each language selected. Each .EXE file will create a subdirectory labeled with the language name attached to the .EXE. These subdirectories will also be in the directory from which TSETUP is run, which is the same directory that contains the subdirectory TRANSEXEXE.

Within each resulting language subdirectory will reside the necessary files for that language. Epi Info 2000 can then access the files in each language subdirectory to enable translation to that language.

A translation of TSETUP itself can be provided by translators for end users, if desired.

ARJ

To create a self-expanding .EXE with ARJ, do the following:

First, ensure that a subdirectory named TRANSEXEXE exists in the directory where TSETUP resides.

Next, place all non-compressed directories and files that you wish to add to the self-expanding .EXE into a subdirectory analogous to the following:

c:\source\Spanish

(Spanish is used for example: any language name of eight characters or less is acceptable.)

Then, from the MS-DOS prompt, type the following commands:

```
cd c:\source\Spanish
```

```
c:\epi2000\ARJ a -r c:\epi2000\transexex\Spanish.exe *.* -je
```

where

c:\source\Spanish

is the subdirectory where non-compressed source files and subdirectories reside. These files and subdirectories will be incorporated into the resulting compressed, self-expanding .EXE file (in this case called "Spanish.exe" under c:\epi2000\transexex).

ARJ

is the compression program, which should reside in the c:\epi2000 directory. In order to run the program from within the c:\source\Spanish subdirectory, the path to ARJ, c:\epi2000\ARJ, must be specified.

a

is a parameter that instructs ARJ to "add" files to the self-expanding .EXE.

-r

is a parameter that instructs ARJ to include subdirectories, if encountered in the source subdirectory ("r" stands for "recurse directories").

c:\epi2000\transexex\Spanish.exe

is the filename for the self-expanding .EXE file that will be created.

.

will include ALL files (and directories) resident in the specified source subdirectory.

-je

is a parameter that causes ARJ to create a self-expanding compressed file with the suffix .EXE.

As previously stated, the language “Spanish” has been used for example. Any language name, up to eight characters, may be substituted. TSETUP is not sensitive to letter case, so “Spanish” will work as well as “spanish” or “SPANISH.”

In order to install multiple languages, simply repeat the above procedure for each language, using appropriate naming conventions for each source subdirectory (e.g., c:\source\French, c:\source\Swedish) and EXE file (e.g., c:\epi2000\transexex\French.exe, c:\epi2000\transexex\Swedish.exe). A self-expanding .EXE file will be created in the c:\epi2000\transexex subdirectory for each language, and the TSETUP program can then read and process each of those .EXEs constructed, so that multiple languages can be installed during one run of TSETUP, as long as a .EXE file exists (in the TRANSEXEX subdirectory) for each language.

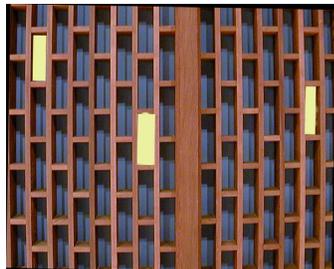
Each time TSETUP runs, if, in the destination subdirectory, files already exist for a specific language, those files will be overwritten by contents of the .EXE file for that language.

Please consult the documentation for ARJ, entitled ARJ.txt (provided), for further assistance.

Fonts, Unicode, Right-to-Left, and other Features Required for Non-European Languages

Epi Info is programmed to set its default font to that of the computer on which the program is running. The Visual Basic in which it is programmed is capable of displaying Unicode, the two-character code used for Chinese and other languages for which ASCII is not adequate. Arabic, Hebrew, and other right-to-left languages should display properly on computers that have this feature included in the Windows operating system.

Statistics



What Is the Best Estimate? Can the Observations be Explained by Random Variation?

Overview

Selection of statistical methods depends on the type of data and the purpose of the analysis. Epi Info provides a number of statistical methods within the commands **FREQ**, **TABLES**, **MEANS**, and **REGRESS**. Advanced statistical techniques are discussed in separate chapters (see chapters on Logistic Regression and Kaplan-Meier Survival analysis).

The general outline of this chapter is to describe a command, show its output, and then provide interpretation. Formulas are given for reference purposes, but can be considered optional reading.

The commands used for statistical analysis in Epi Info, the types of data to be analyzed, and the purposes of various analyses are as follows:

For all types of data	
Purpose	See the data, record by record
Analysis command	LIST

For Text, Numbers, or Dates—Categorical Data	
Purpose	Determine the frequency of each value
Analysis command	FREQ
Statistical method(s)	Confidence intervals on proportions

For Text, Numbers, or Dates—Categorical Data—2 values per variable	
Purpose	Evaluate Associations between Variables Cross tabulations 2x2 tables
Analysis command	TABLES (Risk Factor) (Outcome)
Statistical method(s)	Odds ratios with confidence limits

	Risk Ratio with confidence limits Chi Square Fisher Exact
--	---

For Text, Numbers, or Dates—Categorical Data—2 values per variable with multiple values for a possible confounder or modifier used for stratification	
Purpose	Assessing Confounding and Interaction (Effect Modification)
Analysis command	TABLES (Risk Factor) (Outcome) (Stratifier(s))
Statistical method(s)	Mantel-Haenszel odds ratio with confidence limits Summary Risk Ratio with confidence limits Fisher Exact

For Text, Numbers, or Dates—Categorical Data—More than 2 Values per Variable	
Purpose	Cross tabulations R x C
Analysis command	TABLES (Risk Factor) (Outcome) optional (Stratifier(s))
Statistical method(s)	Chi Square

For Text, Numbers, or Dates—Categorical Data with Matched Cases and Controls	
Purpose	Cross tabulations with Stratification 2 x 2 x M
Analysis command	TABLES (Risk Factor) (Outcome) (MatchVariable)
Statistical method(s)	Mantel-Haenszel odds ratio with confidence limits Risk Ratio with confidence limits Fisher Exact

For Numbers—Continuous Data, Single Variable	
Purpose	Express series of numbers as a single value
Analysis command	MEANS VariableName
Statistical method(s)	Sum, Mean, Median, Mode, Percentiles, Standard Deviation, Variance

For Numbers—Continuous Data, Single Variable	
Purpose	Test whether a series of differences (e.g., before and after) differs from zero
Analysis command	MEANS VariableName
Statistical method(s)	t-test

For Numbers—Continuous Data, Grouped by Another (Categorical) Variable	
Purpose	See if Group Values Differ
Analysis command	MEANS Number GroupVar Bartlett's test for homogeneity of variances
Statistical method(s)	ANOVA (and Student's t-test if two groups) Kruskal-Wallis Test (Mann-Whitney/Wilcoxon if two groups)

For Numbers—Continuous Data, Two Numeric Variables	
Purpose	Measure Correlation
Analysis command	REGRESS Outcome=Risk Factor
Statistical method(s)	Pearson Correlation Coefficient Mean, Beta coefficient, Lower and Upper Confidence Limits, Standard Error, F-Test, Y-Intercept

For Numbers—Continuous Data, More Than Two Numeric Variables	
Purpose	Predict Outcome from Risk Factors

<i>Analysis Command</i>	REGRESS Outcome=Risk Factor(s)
<i>Statistical method(s)</i>	Pearson Correlation Coefficient Means, Beta coefficients, Lower and Upper Confidence Limits, Standard Error, Partial F-Test, Y- Intercept

FREQ, TABLES, and MEANS can be used for either individual or summary data. For summary data, the name of a field containing Count data should be specified in the Weight or Count box in the command dialog. This feature can also be used for weighting statistical data—to compensate for non-proportional sampling, for example.

For the FREQ, MEANS, and REGRESS commands, nomenclature and formulas are based primarily on the text by Rosner (1995). For the TABLES command, the nomenclature and formulas are primarily from Kleinbaum, Kupper, and Morgenstern (1982) with some modification.

The commands assume that the data were collected using simple random (or unbiased systematic) sampling. If the data were collected using a more complex design, such as cluster sampling, then methods used to analyze the data need to take the design into account. Epi Info for DOS, available from the Epi Info web site mentioned on the title page of this manual, provides the **CSample** program for the analysis of data collected using complex designs, and this can be linked to the Epi Info 2000 menu if desired for use after exporting data to an Epi Info 6 file.

This chapter provides notes on the statistical output. Although the programs have been checked against manual calculations and other computer programs to a certain extent, this is no guarantee of their accuracy under all conditions. A result that does not seem reasonable should be checked against other computer programs or with a statistician.

Computers can produce meaningless results almost *ad infinitum* (well, nearly), if applied in an inappropriate way. Statistical significance must be considered along with biological plausibility and importance (i.e., public health significance or clinical significance). Huge samples can make hairline differences “statistically significant,” and small samples can fail to show “significant” results for potentially important findings. It is best to seek expert epidemiological or statistical advice when there is doubt about the appropriateness or interpretation of a statistical analysis or test.

Background

The majority of statistical and epidemiological analyses performed by Epi Info are based on the assumption that the data were collected from an infinitely large population using simple random sampling. Point estimates, such as proportions, risk ratios, odds ratios, means, etc., are sample estimates.

There is controversy concerning the use of p-values versus confidence intervals (see Rothman, 1986). Epi Info generally provides both. Another area of controversy is the use one-sided versus two-sided p-values. In the majority of situations, the two-tailed is appropriate, but there are situations where one-tailed could be appropriate (see Fleiss, 1981). For some chi square tests, the uncorrected and corrected values are provided; statisticians disagree over the appropriateness of the correction.

Choosing an Epi Info Command to Analyze Common Types of Data

For All Types of Data—The LIST Command

The simplest and sometimes the best way to analyze data is to produce a line listing, using the LIST command. LIST has options to determine which variables are listed, divide the columns into page-sized groups, and display or not display line numbers. Further details are given in the Commands chapter. By using the SELECT command, the records displayed can be limited to those with particular values for one or more variables. Records marked as deleted can be displayed or not, depending on a setting in the Options dialog.

For Text, Numbers, and Dates—The FREQ Command

The FREQ command is used to determine the frequency of values for numeric, character, or date variables. The output shows a table of the values for a variable, the number and percent of records having each value, and the confidence intervals for each value as a proportion of the total. For numeric data only, descriptive statistics such as mean and standard deviation are also shown. In the example below, the OSWEGO view is used with the command FREQ ILL.

The commands

```
READ viewOSWEGO
FREQ ILL
```

produce the following output, which is explained below:

Current View
No Of Records

C:\data\Sample.mdb : viewOswego
75 *Date*

10/26/99 8:26:22 PM

Frequencies of *ILL*?

ILL

ILL	Frequency	Percent	Cum Percent
Yes	46	61.3%	61.3%
No	29	38.7%	100.0%
Total	75	100.0%	100.0%

95% Conf Limits

Yes	49.4%	72.4%
No	27.6%	50.6%

The output has a title showing the View or Data table from which the information was taken, the number of records currently selected, and selection criteria, if any.

In the table:

- **ILL** shows values for the variable ILL. The representation of Yes and No is determined by a SET command under Options, but the underlying data values are always 0 for No and 1 for Yes.
- **Frequency** is the number of records in the dataset having the indicated value. In the example, there are 46 people who said they were ill and 29 who were not ill.
- **Percent** is the number of ill divided by the total (i.e., 46/75 or 61.3%).
- **Cum. Percent** The cumulative percent cumulatively adds the percent column.
- **Total** shows the total number of observations in table (in this example, 75) and the total percent, which is always 100.0%
- **95% Confidence Limits** are given for each proportion. If the 75 records had been chosen randomly from a large number of interviews, we would predict with 95% confidence (i.e., be wrong 5 out of 100 tries) that the number of ill in the larger population lies between 49.4% and 72.4%. In general, a larger sample brings the confidence limits closer together. In this case, all available attendees at the church supper were interviewed, and the meaning of the confidence intervals is somewhat different.

For Summary Data With a COUNT Field—FREQ with a Weight Variable

Many datasets are in aggregate or summary form, much like the frequency discussed above. Individual records contain a value for a variable such as AGE, but each record also contains another field, such as COUNT, that indicates the number of individuals or records associated with the value. This COUNT or WEIGHT variable is used to determine the number of

individual instances of the other fields in the record. Thus a record with AGE=10 and COUNT=2 is taken to represent 2 10-year-olds. If another record has AGE=10 and COUNT=5, then the total number of 10-year-olds is 7. Using the FREQ command with such a dataset produces a list of variable values quite similar to a LISTing of the records, but the following commands and output show how the results differ from a LIST.

```
READ 'C:\data\Sample.mdb':viewAgeWithCount
FREQ Age WEIGHTVAR=Count
```

Current View C:\data\AgeSummary.MDB : viewAgeSummary
 No Of Records 10 Date 10/24/99 9:16:03 PM

Frequencies of Age

AGE , Weight = AGESUMMARY.COUNT

Age	Frequency	Percent	Cum Percent
1	5	5.9%	5.9%
2	11	12.9%	18.8%
3	19	22.4%	41.2%
4	17	20.0%	61.2%
5	20	23.5%	84.7%
6	6	7.1%	91.8%
7	3	3.5%	95.3%
8	2	2.4%	97.6%
9	1	1.2%	98.8%
10	1	1.2%	100.0%
Total	85	100.0%	100.0%

95% Conf Limits

1	1.9%	13.2%
2	6.6%	22.0%
3	14.0%	32.7%
4	12.1%	30.1%
5	15.0%	34.0%
6	2.6%	14.7%
7	0.7%	10.0%
8	0.3%	8.2%
9	0.0%	6.4%
10	0.0%	6.4%

The Frequency column is taken from the value of the COUNT variable, and the results resemble FREQ AGE in the table of individual records from which the summary table originally came.

Summary tables are produced in Epi Info from individual records by specifying an OUTTABLE name in a FREQUENCY command. Summary results are sent to a data table named by the user, and are then ready for retrieval with the FREQ or TABLES command and a COUNT variable.

The total number of observations (n) is based *not* on the number of records in the file (in this example, 10 records), but on the sum of those in the Freq column (denoted as f_i in equations), in this example, 85.

The ability to process summary data is useful in public health surveillance systems, where a number of local agencies may process data to produce summary tables and then send the tables to a central agency where they can be combined into a large table that is in turn processed by running FREQ (Variable of Interest) WEIGHTVAR=COUNT.

Formulas for Confidence Limits of Proportions

For small sample sizes, the confidence intervals are calculated using the exact, or exact mid-p binomial, method. For larger samples sizes, the Fleiss quadratic method is used. The confidence interval formulas are:

Fleiss quadratic, lower bound:

$$\frac{(2n\hat{p} + Z_{1-\alpha/2}^2 - 1) - Z_{1-\alpha/2} \sqrt{Z_{1-\alpha/2}^2 - (2 + 1/n) + 4\hat{p}(n\hat{q} + 1)}}{2(n + Z_{1-\alpha/2}^2)}$$

Fleiss quadratic, upper bound:

$$\frac{(2n\hat{p} + Z_{1-\alpha/2}^2 + 1) + Z_{1-\alpha/2} \sqrt{Z_{1-\alpha/2}^2 + (2 - 1/n) + 4\hat{p}(n\hat{q} - 1)}}{2(n + Z_{1-\alpha/2}^2)}$$

Where:

n = the denominator

\hat{p} = the proportion

$Z_{1-\alpha/2}$ = the two-sided Z value, 1.96 for a 95% confidence interval

$\hat{q} = \hat{p} - 1$

For the Fisher exact limit, the lower bound (Rothman) is:

$$\alpha / 2 = \sum_{k=a}^n \binom{n}{k} p_{LB}^k (1 - p_{LB})^{n-k}$$

and the upper bound:

$$\alpha / 2 = \sum_{k=0}^a \binom{n}{k} p_{UB}^k (1 - p_{UB})^{n-k}$$

where

α = the tail probability, e.g., 0.05 for a 95% confidence interval

a = the numerator

p_{LB} = the lower bound proportion

p_{UB} = the upper bound proportion

For the mid-p exact, the lower bound (Rothman) is:

$$\alpha / 2 = \frac{1}{2} \binom{n}{a} p_{LB}^a (1 - p_{LB})^{n-a} + \sum_{k=a+1}^n \binom{n}{k} p_{LB}^k (1 - p_{LB})^{n-k}$$

and the upper bound:

$$\alpha / 2 = \frac{1}{2} \binom{n}{k} p_{UB}^a (1 - p_{UB})^{n-a} + \sum_{k=0}^{a-1} \binom{n}{k} p_{UB}^k (1 - p_{UB})^{n-k}$$

For the exact methods, Epi Info uses a fast method to perform these calculations based on the F-distribution. The lower and upper F-distribution methods for calculating Fisher exact confidence limits are:

$$\sum_{k=a}^n \Pr(k) = \Pr \left[F_{2(b+1), 2a} > \frac{2(b+1)(1 - p_{lb})}{2ap_{lb}} \right]$$

$$\sum_{k=0}^a \Pr(k) = 1 - \Pr \left[F_{2b, 2(a+1)} > \frac{2b(1-p_{ub})}{2(a+1)p_{ub}} \right]$$

where

$$b = n - a$$

Lower and upper F-distribution method for calculating mid-p exact confidence limits:

$$\sum_{k=a}^n \Pr(k) = \Pr \left[F_{2(b+1), 2a} > \frac{2(b+1)(1-p_{lb})}{2ap_{lb}} \right] - \frac{1}{2} \binom{n}{a} p_{LB}^a (1-p_{LB})^{n-a}$$

$$\sum_{k=0}^a \Pr(k) = 1 - \Pr \left[F_{2b, 2(a+1)} > \frac{2b(1-p_{ub})}{2(a+1)p_{ub}} \right] - \frac{1}{2} \binom{n}{k} p_{UB}^a (1-p_{UB})^{n-a}$$

Epi Info also provides a number of useful graphics for displaying frequency data. When there are relatively few levels (around 10 or less) and the data are nominal (“Hepatitis A,” “Hepatitis B,” “Hepatitis C”) or ordinal (1, 5, 6), the PIE chart and BAR chart may be used to present the data graphically. If the data are continuous or date fields, the LINE and HISTOGRAM charts may be useful.

For Text, Numbers, or Dates—Cross Tabulations with the TABLES Command

The TABLES command is used to create a table of categorical data, often called a cross-tabulation. A single table of 2 or more rows and 2 or more columns can be provided, which will be described as an R x C table (Row by Column). In addition to the row and column variables, the user can stratify on additional variables, which will be described as an R x C x S table (Row by Column by Strata).

The general form of the TABLES command is:

```
TABLES <row var> <column var> {<stratifying vars>}
```

or, for epidemiologic data with specific exposure and disease variables:

```
TABLES <exposure var> <disease var> {<stratifying vars>}
```

More information on the syntax for the TABLES command can be found in the Commands chapter.

Yes/No or True/False Variables and 2x2 Tables

In epidemiology, 2x2 tables are frequently used. In these tables, there is usually an “exposure” variable that has two levels (e.g., exposed vs. not exposed) and a dichotomous (2-value) outcome variable (e.g., the person had the disease or outcome of interest or they did not; see Table 1). With 2x2 tables, the odds ratio (OR), risk ratio (RR), and other parameters can be calculated.

To get the correct estimates for these parameters in Epi Info, the table must be set up as follows:

Table 1. Standard table setup and notation for Epi Info (count data and unmatched data)

Exposed	Disease		Total
	Yes	No	
Yes	a	b	n ₁
No	c	d	n ₀
Total	m ₁	m ₀	n

The “exposure” is the row variable, and the “disease” or “outcome” is the column variable. The cases with the exposure of interest should be in the first row (in this example “Yes”) and those without the exposure in the second row. For the “disease” or “outcome” variable, those with the outcome of interest should be in the first column (in this example “Yes”) and those without the disease in the second column.

Note that the “standard” table setup may differ between textbooks. Epi Info uses the table setup (as shown in Table 1) used in the texts by Rosner (1995), Schlesselman (1982), Kahn and Sempos (1989), Kelsey et al. (1986), and others. Some texts have the disease variable as the row variable and exposure as the column variable, such as Kleinbaum et al. (1982), Rothman (1986), Greenberg et al. (1996), and others.

Parameters, such as the risk ratio and odds ratio, are calculated regardless of whether the table is set up correctly or incorrectly. The computer program relies on the user to assure that the information is provided correctly, as depicted in Table 1. What happens if the table is set up incorrectly? There are eight possible ways to mix up a 2x2 table. Only two possible odds ratios can be calculated, the true OR (1.75) or its inverse (0.57 or 1/1.75). There are eight different possible “risk ratios” that can be calculated, of which only one (RR=1.24) is correct.

As an example of a 2x2 table, the example below uses a dataset from Kleinbaum, Kupper, and Morgenstern, the Evans County heart disease study in white males after seven years of follow-up. In the output below the exposure is a dichotomous variable, CAT (catecholamine level); those exposed have high CAT levels (1) and the nonexposed have low CAT levels (0). The

outcome is coronary heart disease (CHD); those who had a coronary event are 1 and those without CHD are 0. The first and last three records of the data for CAT and CHD are depicted below.

Evans County data listing for CAT and CHD

Current View **C:\data\Data.mdb : viewKkmyn**
No Of Records **609** *Date* **10/21/1999 4:51:35 PM**

ID	CAT	CHD
21	0	0
31	0	0
51	1	1
19091	0	0
19121	0	0
19161	0	0

The output from the TABLES command for a 2x2 table is provided in three sections. The first section provides the table, the second the parameter estimates, and the third the statistical tests.

TABLES CAT CHD

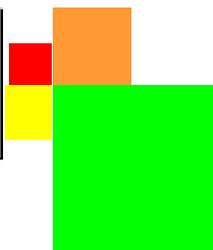
First section, the Table. We continue with the above example, showing the TABLES command for a 2x2 Table:

Current View **C:\data\Sample.mdb : viewEvansCounty**
No Of Records **609** *Date* **10/26/99 8:26:25 PM**

CAT by CHD

CHD

CAT	Yes	No	Total
Yes	27	95	122
No	44	443	487
Total	71	538	609



Single Table Analysis

	Point Estimate	95% Confidence Interval		
		Lower	Upper	
PARAMETERS: Odds-based				
Odds Ratio (cross product)	2.8615	1.6878	4.8514	(T)
Odds Ratio (MLE)	2.8554	1.6690	4.8350	(M)
		1.6148	4.9853	(F)
PARAMETERS: Risk-based				
Risk Ratio (RR)	2.4495	1.5837	3.7887	(T)
Risk Difference (RD)	13.0962	5.3021	20.8903	(T)

(T=Taylor series; C=Cornfield; M=Mid-P; F=Fisher Exact)

STATISTICAL TESTS	Chi-square	1-tailed p	2-tailed p
Chi square - uncorrected	16.2465		0.0000567826
Chi square - Mantel-Haenszel	16.2198		0.0000575712
Chi square - corrected (Yates)	14.9998		0.0001086935
Mid-p exact		0.0000910000	
Fisher exact		0.0001400000	

When the table is set up correctly for Epi Info (as depicted in Table 1), the row proportions have epidemiologic meaning. In a cohort study, in the first row, the row percentage in the first column is the “risk” of disease among the exposed; in the second row, the risk in the unexposed. If the data are from an outbreak of acute disease, these are sometimes referred to as “attack rates.” If the data are based on prevalence information, the row proportions are the prevalence of disease in the exposed and unexposed.

This next section provides the equations used for calculating the point estimates and their confidence limits. Users not interested in the equations can skip beyond them to the section on interpretation of the various parameters.

Equations for Parameter Estimates and Confidence Intervals for 2x2 Tables

The odds-based parameters in the output are used for unmatched case-control studies, but can also be used for cohort or cross-sectional data. If the cases with disease are incident cases, sometimes the odds ratio is called the “risk odds ratio” (**ROR**); if the cases with disease are based on prevalent cases, the odds ratio may be referred to as the “prevalence odds ratio” (**POR**). The following two “odds” could be calculated (but are not shown in the output):

Odds of exposure among the cases (O_{ca}) = a/c

Odds of exposure among the controls (O_{co}) = b/d

The odds-based parameters are calculated as:

Odds Ratio (**OR**) =

$$O_{ca}/O_{co} =$$

$$ad / bc$$

(Eq 26)

Frequently Equation 26 is referred to as the “cross product.” Note that Equation 26 would also be the result of the odds of disease among the exposed, relative to the odds of disease among the nonexposed. Another estimate of the odds ratio provided is called the maximum likelihood estimate (MLE). The method by which it is computed is described in Martin and Austin (1991). In most cases, its value is very close to that of the traditional cross-product OR.

The confidence interval for the OR is calculated based on the Taylor series approach as:

$$\text{ORexp} \left[\pm Z_{1-\alpha/2} \sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}} \right]$$

The lower bound of the confidence interval for the OR based on Cornfield’s method is calculated as the smallest real root of the equation (KKM):

$$(a - a_L - 0.5)^2 \left(\frac{1}{a_L} + \frac{1}{m_1 - a_L} + \frac{1}{n_1 - a_L} + \frac{1}{n_0 - m_1 + a_L} \right) = Z_{1-\alpha/2}^2$$

where a , m_1 , n_0 , n_1 are as depicted in Table 1, and a_L is a value that is iteratively changed in value until the above equation equals the Z value. The value a_L is subject to the maximum and minimum condition:

$$\max(0, m_1 - n_0) \leq a_L \leq \min(n_1, m_1)$$

Once the value a_L is determined, the lower bound of the confidence interval is calculated as:

$$OR_L = a_L (n_0 - m_1 + a_L) / (m_1 - a_L)(n_1 - a_L)$$

The upper bound of the Cornfield confidence limit is calculated as the largest real root of the following equation:

$$(a - a_U + 0.5)^2 \left(\frac{1}{a_U} + \frac{1}{m_1 - a_U} + \frac{1}{n_1 - a_U} + \frac{1}{n_0 - m_1 + a_U} \right) = Z_{1-\alpha/2}^2$$

The value a_U is subject to the maximum and minimum condition:

$$\max(0, m_1 - n_0) \leq a_U \leq \min(n_1, m_1)$$

and the upper bound of the confidence interval is calculated as:

$$OR_U = a_U (n_0 - m_1 + a_U) / (m_1 - a_U)(n_1 - a_U)$$

The lower bound of the Fisher exact confidence interval is calculated as:

$$\alpha / 2 = \frac{\sum_{k=a}^{\min(n_1, m_1)} \binom{n_1}{k} \binom{n_0}{m_1 - k} OR^k}{\sum_{k=\max(0, m_1 - n_0)}^{\min(n_1, m_1)} \binom{n_1}{k} \binom{n_0}{m_1 - k} OR^k}$$

and the upper bound of the Fisher exact confidence interval as:

$$1 - \alpha / 2 = \frac{\sum_{k=a+1}^{\min(n_1, m_1)} \binom{n_1}{k} \binom{n_0}{m_1 - k} OR^k}{\sum_{k=\max(0, m_1 - n_0)}^{\min(n_1, m_1)} \binom{n_1}{k} \binom{n_0}{m_1 - k} OR^k}$$

Lower and upper bounds for the exact mid-p confidence interval are calculated by:

$$\alpha / 2 = \frac{\frac{1}{2} \binom{n_1}{a} \binom{n_0}{b} OR^k + \sum_{k=a}^{\min(n_1, m_1)} \binom{n_1}{k} \binom{n_0}{m_1 - k} OR^k}{\sum_{k=\max(0, m_1 - n_0)}^{\min(n_1, m_1)} \binom{n_1}{k} \binom{n_0}{m_1 - k} OR^k}$$

and

$$1 - \alpha / 2 = \frac{\frac{1}{2} \binom{n_1}{a} \binom{n_0}{b} OR^k + \sum_{k=a+1}^{\min(n_1, m_1)} \binom{n_1}{k} \binom{n_0}{m_1 - k} OR^k}{\sum_{k=\max(0, m_1 - n_0)}^{\min(n_1, m_1)} \binom{n_1}{k} \binom{n_0}{m_1 - k} OR^k}$$

respectively. Fast iterative solutions to the exact confidence intervals and exact p-values are calculated using a program by Martin and Austin (1991).

Risk Ratio, frequently called “relative risk”:

$$(\mathbf{RR}) = Re / Ru$$

Confidence interval for the RR based on the Taylor series approach:

$$RR \exp \left[\pm Z_{1 - \alpha / 2} \sqrt{\frac{1 - Re}{n_1 Re} + \frac{1 - Ru}{n_0 Ru}} \right]$$

$$\text{Risk Difference } (\mathbf{RD}) = Re - Ru$$

Confidence interval for the RD based on the Taylor series approach:

$$RD \pm Z_{1 - \alpha / 2} \sqrt{\frac{Re(1 - Re)}{n_1} + \frac{Ru(1 - Ru)}{n_0}}$$

Calculations and Interpretation of Parameter Estimates for a 2x2 Table

Using the Evans County CAT/CHD example, the parameters are calculated as:

OR = $44 * 95 / (443 * 27) = 2.86$. Interpretation: The odds of CHD in those with high CAT levels is around 2.9 times greater than the odds of CHD among those with low CAT levels. Because the frequency of disease is relatively high (i.e., $R = 12\%$), the OR overestimates the RR ($RR=2.45$).

Re = $27/122 = 0.2213$ or 22.13%

Ru = $44/487 = 0.0905$ or 9.05%

R = $71/609 = 0.1166$ or 11.66%

The interpretation would be that 22.1% of men with high CAT levels developed CHD, compared with 9.0% among those with low CAT levels. The overall risk of CHD was 11.7%, and 20% had high CAT levels.

If the data are based on prevalence data, such as from a cross-sectional study, the prevalence in the exposed and nonexposed could be calculated the same as the risks. In the rest of this chapter, although the term “risks” will be used, if the data are based on prevalence, substitute the term “prevalence” for “risk.” For example, rather than stating a “risk ratio” was calculated, use the term “prevalence ratio.” Unmatched case-control studies should not use any of the risk-based parameters described below.

Continuing with the Evans County CAT/CHD example:

RR = $22.13\% / 9.04\% = 2.45$. Interpretation: The risk of CHD in men with high CAT levels is around 2.5 times greater than the risk in those with low CAT.

RD = $22.13\% - 9.03\% = 13.10\%$. Interpretation: The absolute risk of CHD in men with high CAT levels is around 13% higher than the risk in those with low CAT levels.

Equations for Statistical Tests for a 2x2 Table

The uncorrected chi square is calculated as

$$\chi_1^2 = \frac{(n)(ad - bc)^2}{n_1 n_0 m_1 m_0}$$

and the Mantel-Haenszel chi square as

$$\chi_1^2 = \frac{(n-1)(ad - bc)^2}{n_1 n_0 m_1 m_0}$$

The Yates corrected chi square as

$$\chi_1^2 = \frac{(n-1)(|ad - bc| - \frac{n}{2})^2}{n_1 n_0 m_1 m_0}$$

Mid-p exact, when $a < m_1 n_1 / n$:

$$p = \left(\frac{1}{2}\right) \frac{\binom{n_1}{a} \binom{n_0}{m_1 - a}}{\binom{n}{m_1}} \sum_{k=\max(0, m_1 - n_0)}^{a-1} \frac{\binom{n_1}{k} \binom{n_0}{m_1 - k}}{\binom{n}{m_1}}$$

and when $a > m_1 n_1 / n$

$$p = \left(\frac{1}{2}\right) \frac{\binom{n_1}{a} \binom{n_0}{m_1 - a}}{\binom{n}{m_1}} \sum_{k=a+1}^{\min(n_1, m_1)} \frac{\binom{n_1}{k} \binom{n_0}{m_1 - k}}{\binom{n}{m_1}}$$

Fisher exact, when $a < m_1 n_1 / n$:

$$p = \sum_{k=\max(0, m_1-n_0)}^a \frac{\binom{n_1}{k} \binom{n_0}{m_1-k}}{\binom{n}{m_1}}$$

and when $a > m_1 n_1/n$:

$$p = \sum_{k=a}^{\min(n_1, m_1)} \frac{\binom{n_1}{k} \binom{n_0}{m_1-k}}{\binom{n}{m_1}}$$

Interpretation of Statistical Tests

All of the p-values in the example are < 0.001 , indicating a statistically significant association between CAT and CHD in the study population. One question is the use of one-tailed vs. two-tailed p-values. Many authors argue that two-tailed p-values are appropriate in the majority of situations (for a discussion of this issue, see Fleiss, 1981). Which of the two-sided p-values should be used? When the cell sizes are reasonably large, all of the p-values will be similar; when the data are sparse, the exact p-values, with the exact mid-p p-values, seem to be the most frequently recommended. Prior to performing analyses, the user should decide which p-value(s) to use to determine statistical significance. Occasionally one method will have a statistically significant p-value and others may not.

Assessing Confounding and Interaction by Stratification

2 X 2 X S TABLES

A useful technique for analysis in epidemiology is “stratification,” meaning that the data are broken into groups or strata based on one or more variables other than the two being used in 2x2 tables. In this situation there is an “exposure” variable of two levels (e.g., exposed vs. not exposed), an outcome variable with two levels (e.g., the person had the disease or outcome of interest or they did not), and a stratifying variable, which may have two or more levels. Stratification is performed to investigate whether a stratifying variable is an effect modifier of the exposure-disease relation, a confounder of the relation, or neither. The output provided with stratification includes a table, parameter estimation, and statistics for each level of the stratifying

variable; after all the stratum-specific results are provided, the *summary* or *pooled* results are provided along with tests for interaction.

An example is provided below using the Evans County data with the command

```
TABLES CAT CHD AGEG1
```

The stratifying variable AGEG1 is age recoded as “Yes” for those ≥ 55 years of age and “N” for those < 55 years of age. The format of the output is a 2x2 table of CAT by CHD at each level of the stratifying variable, followed by information that summarizes the stratified data. At the top of the output is a menu item to let the user know that a stratified analysis has been performed. The first stratification table shown in the Evans County stratification example is the relation between CAT and CHD *among older individuals only* (AGEG1 = “Yes”).

Example: Stratified analyses, dichotomous exposure and disease variables, Evans County Data.

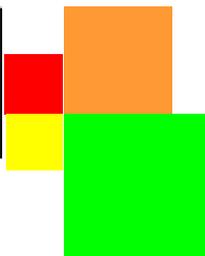
```
Current View          C:\data\Data.mdb : viewEvansCounty
No Of Records        609                               Date          10/21/1999 5:03:33 PM
```

CAT by CHD, stratified by AGEG1

AGEG1 = 1

CHD

CAT	1	0	Total
1	23	74	97
0	20	134	154
Total	43	208	251



Single Table Analysis

	Point Estimate	95% Confidence Interval	
		Lower	Upper
PARAMETERS: Odds-based			
Odds Ratio (cross product)	2.0824	1.0730	4.0415 (T)
Odds Ratio (MLE)	2.0761	1.0653	4.0719 (M)
		1.0158	4.2778 (F)
PARAMETERS: Risk-based			
Risk Ratio (RR)	1.8258	1.0611	3.1416 (T)
Risk Difference (RD)	10.7243	0.7329	20.7158 (T)

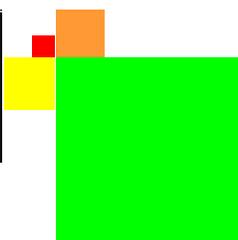
(T=Taylor series; C=Cornfield; M=Mid-P; F=Fisher Exact)

STATISTICAL TESTS	Chi-square	1-tailed p	2-tailed p
Chi square - uncorrected	4.8214		0.0281094006
Chi square - Mantel-Haenszel	4.8022		0.0284244563
Chi square - corrected (Yates)	4.0956		0.0429963635
Mid-p exact		0.0160000000	
Fisher exact		0.0224000000	

AGEG1 = 0

CHD

CAT	1	0	Total
1	4	21	25
0	24	309	333
Total	28	330	358



Single Table Analysis

PARAMETERS: Odds-based	Point Estimate	95% Confidence Interval	
		Lower	Upper
Odds Ratio (cross product)	2.4524	0.7788	7.7226 (T)
Odds Ratio (MLE)	2.4440	0.6705	7.3447 (M)
		0.5647	8.1064 (F)
PARAMETERS: Risk-based			
Risk Ratio (RR)	2.2200	0.8354	5.8996 (T)
Risk Difference (RD)	8.7928	-5.8441	23.4297 (T)

(T=Taylor series; C=Cornfield; M=Mid-P; F=Fisher Exact)

STATISTICAL TESTS	Chi-square	1-tailed p	2-tailed p
Chi square - uncorrected	2.4937		0.1143017715
Chi square - Mantel-Haenszel	2.4868		0.1148087684
Chi square - corrected (Yates)	1.4232		0.2328723203
Mid-p exact		0.0778000000	
Fisher exact		0.1199000000	

SUMMARY INFORMATION

Parameters	Point	95% Confidence Interval	
	Estimate	Lower	Upper
Odds Ratio Estimates			
Crude OR (cross product)	2.8615	1.6878,	4.8514 (T)
Crude OR (MLE)	2.8554	1.6690,	4.8350 (M)
		1.6148,	4.9853 (F)
Adjusted OR (MH)	2.154	1.3218,	3.5093 (R)
Adjusted OR (MLE)	2.1607	1.2017,	3.8588 (M)
		1.1564,	4.0032 (F)
Risk Ratios (RR)			
Crude Risk Ratio (RR)	2.4495	1.5837,	3.7887
Adjusted RR (MH)	1.8960	1.1773,	3.0535

(T=Taylor series; R=RGB; M=Exact mid-P; F=Fisher exact)

STATISTICAL TESTS (overall assoc)	Chi-square	1-tailed p	2-tailed p
MH Chi square - uncorrected	6.9871		0.0082
MH Chi square - corrected	6.1826		0.0129
Mid-p exact		0.0052	
Fisher exact		0.0074	

Based on the risk ratios, it appears for each of the two age groups that the risk of CHD is two times higher in the high CAT individuals compared with the low CAT individuals. Because AGE1 has only two levels, all the stratum-specific tables have been presented. Summary information will be presented next. First, there will be information on how many individuals are included in the summary information.

If any stratum has a zero margin, all individuals from that stratum are excluded from the summary analyses. In the Evans County Data, there were 609 individuals in the dataset, and no stratum had a zero marginal value. Therefore, all individuals in the dataset will be used in the summary analyses. If analysis is to be performed on a stratum with a zero marginal sum, that stratum can be collapsed (combined) with an adjacent stratum. Finally, parameter estimates are provided for the summary data.

The risk ratio, risk difference, and odds ratios are calculated along with confidence intervals. For each of these parameters, both the “crude” and “adjusted” values are provided. “Crude” values are those in which the stratification is ignored. For example, compare the “Crude” RR, RD, and OR from the Evans County stratification example with the results shown for the Evans County CAT/CHD example, which was based on the TABLES CAT CHD command (i.e., **not** stratifying on AGE1). They are exactly the same. In general, the crude parameter estimates will be the same as the parameter estimates from a TABLES command without the stratifying variable when: 1) There are no individuals with known exposure and disease levels but with a missing value for the stratifying variable(s); and 2) there were no strata with a zero margin. The formulas

used for the crude estimates and their confidence intervals are the same as presented for a single 2x2 table. The next section will present the formulas used for the adjusted parameter estimates and their confidence intervals followed by a section on their interpretation.

Equations for Summary Parameter Estimates, Confidence Intervals, and Statistical Tests for Interaction for Stratified 2x2 Tables

The notation for summary measures needs to be changed to take into account the two or more strata. Table 2 looks similar to Table 1, except for the subscript *i* to denote more than one table and *s* strata.

Standard table setup and notation for stratified data, Epi Info (count data)

Exposed	Disease		Total
	Yes	No	
Yes	a_i	b_i	n_{1i}
No	c_i	d_i	n_{0i}
Total	m_{1i}	m_{0i}	n_i

Three adjusted values are provided for the odds ratio: The Mantel-Haenszel, the direct, and the maximum likelihood methods. First, for the Mantel-Haenszel method, the point estimate is:

$$OR_{MH} = \frac{\sum_{i=1}^s \frac{a_i d_i}{n_i}}{\sum_{i=1}^s \frac{b_i c_i}{n_i}}$$

with a confidence interval of

$$OR_{MH} \exp(\pm Z_{1-\alpha/2} SE[\ln OR_{MH}])$$

where

$$SE[\ln OR_{MH}] = \sqrt{\frac{\sum_{i=1}^s P_i R_i}{2 \left[\sum_{i=1}^s R_i \right]^2} + \frac{\sum_{i=1}^s (P_i S_i + Q_i R_i)}{2 \sum_{i=1}^s R_i \sum_{i=1}^s S_i} + \frac{\sum_{i=1}^s Q_i S_i}{2 \left[\sum_{i=1}^s S_i \right]^2}}$$

$$P_i = (a_i + d_i) / n_i$$

$$Q_i = (b_i + c_i) / n_i$$

$$R_i = a_i d_i / n_i$$

$$S_i = b_i c_i / n_i$$

The directly adjusted odds ratio is based on inverse variance weights derived from Taylor series approximations with the point estimate

$$OR_{Direct} = \exp \left(\frac{\sum_{i=1}^s w_i OR_i}{\sum_{i=1}^s w_i} \right)$$

where

$$w_i = \frac{1}{\frac{1}{a_i} + \frac{1}{b_i} + \frac{1}{c_i} + \frac{1}{d_i}}$$

$$OR_i = \frac{a_i d_i}{b_i c_i}$$

and a confidence interval for the directly adjusted odds ratio of

$$OR_{Direct} \exp \left(\pm \frac{Z_{1-\alpha/2}}{\sqrt{\sum_{i=1}^s w_i}} \right)$$

OR(MLE) estimate

The lower bound of the confidence interval for the OR based on Cornfield's method is similar to that described for a single 2x2 table. An estimate of the OR_L is made, and for each stratum, the a_{iL} value is determined iteratively as described for a single 2x2 table. Once the a_{iL} values are determined for each stratum, the following equation is used:

$$\frac{\left(\left(\sum_{i=1}^s a_i - \sum_{i=1}^s a_{iL} \right) - 0.5 \right)^2}{\sum_{i=1}^s \frac{1}{W_i}} = Z_{1-\alpha/2}^2$$

$$W_i = \frac{1}{a_{iL}} + \frac{1}{m_{1i} - a_{iL}} + \frac{1}{n_{1i} - a_{iL}} + \frac{1}{n_{0i} - m_{1i} + a_{iL}}$$

If the value to the left of the equal sign does not equal that on the right (3.84 for a 95% two-sided confidence interval), another estimate of the OR_L is made and the process repeated. Similarly, for the upper bound estimate OR_U , the same process of estimating a_{iU} values is performed, except that the following equation is used:

$$\frac{\left(\left(\sum_{i=1}^s a_i - \sum_{i=1}^s a_{iL} \right) + 0.5 \right)^2}{\sum_{i=1}^s \frac{1}{W_i}} = Z_{1-\alpha/2}^2$$

with the W_i calculated as indicated for the lower bound, substituting the U subscript for L .

Fisher exact and mid-p exact confidence limits are calculated using algorithms by Martin and Austin (1991).

The adjusted RR using the Mantel-Haenszel method is:

$$RR_{MH} = \frac{\sum_{i=1}^s \frac{a_i n_{oi}}{n_i}}{\sum_{i=1}^s \frac{c_i n_{li}}{n_i}}$$

with a confidence interval based on the Robins-Greenland-Breslow method (RGB)

$$RR_{MH} \exp(\pm Z_{1-\alpha/2} SE[\ln RR_{MH}])$$

where

$$SE[\ln RR_{MH}] = \sqrt{\frac{\sum_{i=1}^s (m_{li} n_{li} n_{oi} - a_i c_i n_i) / n_i^2}{\left[\sum_{i=1}^s \frac{a_i n_{oi}}{n_i} \right] \left[\sum_{i=1}^s \frac{c_i n_{li}}{n_i} \right]}}$$

Adjusted RR using the Direct method (based on the inverse variance weights derived from Taylor series approximation for variance):

$$RR_{Direct} = \exp \left[\frac{\sum_{i=1}^s w_i \ln(RR_i)}{\sum_{i=1}^s w_i} \right]$$

where

$$RR_i = \frac{\frac{a_i}{n_{li}}}{\frac{c_i}{n_{oi}}}$$

$$w_i = \frac{1}{\frac{b_i}{a_i n_{1i}} + \frac{d_i}{c_i n_{0i}}}$$

and the confidence interval for the directly adjusted RR as

$$RR_{Direct} \exp \left(\pm \frac{Z_{1-\alpha/2}}{\sqrt{\sum_{i=1}^s w_i}} \right)$$

The directly adjusted Risk Difference (RD) using inverse variance weights derived from binomial variance estimates is calculated as

$$RD_{Direct} = \frac{\sum_{i=1}^s w_i (RD_i)}{\sum_{i=1}^s w_i}$$

where

$$w_i = \frac{1}{\frac{a_i b_i}{n_{1i}^3} + \frac{c_i d_i}{n_{0i}^3}}$$

$$RD_i = \frac{a_i}{n_{1i}} - \frac{c_i}{n_{0i}}$$

with a confidence interval calculated as

$$RD_{Direct} \pm \frac{Z_{1-\alpha/2}}{\sqrt{\sum_{i=1}^s w_i}}$$

To test for interaction for the odds ratio (OR), the chi square test is calculated as:

$$\chi^2_{s-1} = \sum_{i=1}^S \frac{[\ln(OR_i) - \ln(OR_{Direct})]^2}{Var[\ln(OR_i)]}$$

where the $Var[\ln(OR_i)] = 1/W_i$ from the direct OR calculation. For the risk ratio (RR),

$$\chi^2_{s-1} = \sum_{i=1}^S \frac{[\ln(RR_i) - \ln(RR_{Direct})]^2}{Var[\ln(RR_i)]}$$

where the $Var[\ln(RR_i)] = 1/W_i$ from the direct RR calculation. The test for interaction for the risk difference (RD) is

$$\chi^2_{s-1} = \sum_{i=1}^S \frac{[RD_i - RD_{Direct}]^2}{Var(RD_i)}$$

where the $Var(RD_i) = 1/W_i$ from the direct RD calculation. These tests for interaction are sometimes referred to as the “Breslow-Day” test for interaction.

Interpretation of Parameter Estimates and Statistical Tests for Interaction in Stratified 2x2 Tables

The main reason for stratifying data is to determine whether the stratifying variable modifies or confounds the exposure-disease relationship. The word “modifies” means that the stratifying variable is an “effect modifier” of the exposure-disease relationship or, stated another way, there is an “interaction” between the stratifying variable and the exposure-disease relationship. If there is no interaction, then the issue of confounding is assessed.

To determine if there is interaction, look at the p-value for the test for interaction. For the following discussion, it is assumed that the odds ratio is the parameter of interest. For the odds ratio, the p-value for the test for interaction is 0.81; therefore, there does not appear to be significant interaction. When there is no interaction, the stratum-specific odds ratios will be similar. In the Evans County stratification example, the odds ratio (cross product) for the first stratum was 2.08 and for the second stratum 2.45. When there is a statistically significant

interaction, the stratum-specific odds ratios will be different from one another. In the Evans County stratification example, the interaction p-value was not significant; therefore, the next issue is whether or not the stratifying variable confounds the exposure-disease relationship. To assess confounding, the crude odds ratio is compared with an adjusted odds ratio. For example, compare the crude odds ratio (cross product) with the Mantel-Haenszel adjusted odds ratio [OR(MH)]. The crude OR is 2.86 and the adjusted OR is 2.15. The adjusted OR is always a more valid estimate of the “true” OR than the crude OR; the adjusted OR, however, tends to be less precise (i.e., the confidence intervals tend to be a little wider than the crude OR). There is no statistical test for confounding; the analyst must choose between the more valid but less precise estimate (the adjusted OR) versus the less valid but more precise estimate (the crude OR). Some investigators may choose an arbitrary rule for deciding whether the level of confounding is “important” by comparing the crude and adjusted parameters. For example, the decision may be: if the crude and adjusted parameters differ by more than 5% or 10%, the stratifying variable will be considered a confounder. If the stratifying variable does not modify nor confound an exposure-disease relationship, then it could be ignored in any further analyses.

In the Evans County stratification example, there is large difference between the crude and adjusted odds ratios. Therefore, it appears that age (evaluated as “young” vs. “old”) appears to confound the CAT-CHD relationship.

The steps for evaluating the role of the stratifying variable are:

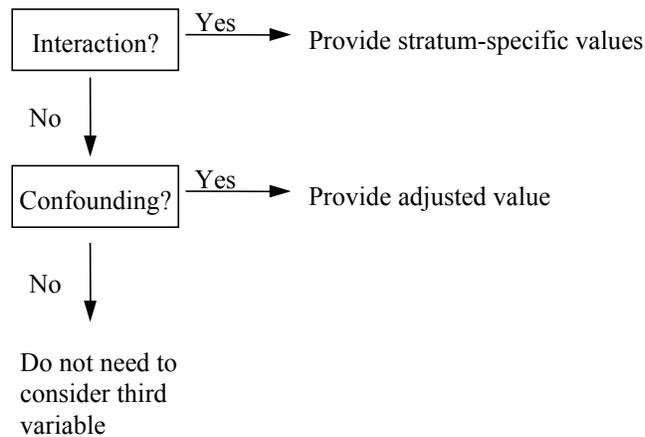
1. Is there interaction? If there is interaction, do *not* use the adjusted or the crude OR. Instead, present the ORs from each stratum.

If there is **no** interaction:

2. Is there confounding? If the crude OR and adjusted OR are similar, then there is no need to stratify on the variable. If the crude OR and adjusted OR are different, use the adjusted OR because it is more valid than the crude OR.

A summary of the process for evaluating interaction and confounding is depicted in Figure 1.

Figure 1. Analytic plan for evaluating stratifying variables



If the parameter of interest is the risk ratio (RR) or risk difference (RD), the same approach for evaluating interaction and confounding is used.

The final section of the summary output for stratified 2x2 tables is composed of the statistical tests. These statistical tests would be used only if it had been decided that the stratifying variable is a confounder of the exposure-disease relationship.

Equations for Statistical Tests for Overall Association in Stratified 2x2 Tables

The *uncorrected* Mantel-Haenszel chi square is calculated as

$$\chi_1^2 = \frac{\left(\sum_{i=1}^s \frac{a_i d_i - b_i c_i}{n_i} \right)^2}{\sum_{i=1}^s \frac{n_{1i} n_{0i} m_{1i} m_{0i}}{(n_i - 1) n_i^2}}$$

and the corrected Mantel-Haenszel chi square as

$$\chi_1^2 = \frac{\left(\left| \sum_{i=1}^s \frac{a_i d_i - b_i c_i}{n_i} \right| - 0.5 \right)^2}{\sum_{i=1}^s \frac{n_{1i} n_{0i} m_{1i} m_{0i}}{(n_i - 1) n_i^2}}$$

Fisher exact, mid-p exact, and the probability that the OR(MLE) = 1 are calculated using algorithms by Martin and Austin (1991).

Interpretation of Statistical Tests for Overall Association in Stratified 2x2 Tables

As stated earlier, the statistical tests for the overall association are used only if there is no interaction but confounding is present. If a p-value < 0.05 is considered statistically significant, then in the Evans County stratification example, in which AGE was shown to confound the CAT-CHD relationship, we would interpret the statistical tests as indicating that the variable CAT is a statistically significant predictor of CHD, controlling for AGE. All of the summary p-values for the Evans County stratification example were < 0.05. Which one should be used? In general, the mid-p exact, two-tailed p-value is the one frequently preferred; if the sample size is large and exact p-values are not provided, then the uncorrected Mantel-Haenszel test is preferred. However, some authors prefer Fisher exact over mid-p exact, and some prefer corrected Mantel-Haenszel over the uncorrected Mantel-Haenszel chi square test. These statistical tests are applicable to the OR, RR, and the RD.

R x 2 and R x 2 x S Tables

Sometimes in epidemiology there is a dichotomous disease variable (disease vs. no disease) and an exposure that takes on more than two levels. For example, a study may be interested in investigating the risk of heart disease among nonsmokers, light smokers, moderate smokers, and heavy smokers. This is usually analyzed by using a test for trend. With smoking, we would expect the risk of heart disease to increase with greater levels of smoking. An R x 2 x S table is one in which a test for trend is stratified across a third variable.

R x C Table (where both R and C are > 2)

When a table has more than 2 rows and 2 columns, *Analysis* will only provide a chi square test.

What To Do After Stratification: Logistic Regression

What is the next step after stratification? Logistic regression is a multivariable model used when the outcome variable is dichotomous (e.g., diseased vs. not diseased). With unmatched data, the unconditional logistic regression model is used. A weakness of stratification is that it may

“spread” the data thin so that in some strata there will be a marginal value of zero; those strata provide no information to the summary results. In addition, with stratification, the exposure and/or stratifying variables can usually take on a limited number of levels (i.e., they are not continuous variables).

Logistic regression allows predictor variables to be either categorical or continuous. However, a potential limitation of most logistic regression programs is that only the odds ratio is estimated; this will be a limitation if the parameter of interest is the risk ratio and the frequency of disease is relatively high (i.e., greater than around 5%), a situation in which the odds ratio overestimates the risk ratio. This is also a limitation if there is interest in the risk difference. Logistic regression is further described in the chapter on that subject.

Matched Cases and Controls—The MATCH Variable in the TABLES Command

The MATCH variable in the TABLES command is used with matched case-control data. With a matched case-control study, it is not generally correct to perform the analyses using 2x2 tables as described above, since this ignores the matching.

A sample dataset included with Epi Info is viewRELY. A matched case-control study was performed on toxic shock syndrome (TSS). Women with TSS had one or more neighborhood controls around the same age. This would result in matching not only on age but on general SES factors. The primary exposure was the use of Rely tampons.

The ID variable is a unique value assigned to each case and one or more matching control(s). The outcome variable is whether the individual is a “case” (usually coded as “Y” or 1) or a “control” (usually coded as “N” or 2). The exposure variable is whether the individual had the exposure, usually coded as “Y” or 1 for those with exposure and “N” or 2 for those without. An example of the first seven cases and controls in the viewRELY dataset is shown below:

Example: Listing of the RELY dataset

REC	IDNUM	ILL	RELY
1	1	Y	Y
2	1	N	N
3	2	Y	Y
4	2	N	N
5	3	Y	Y
6	3	N	N
7	4	Y	Y
8	4	N	N
9	5	Y	N
10	5	N	N
11	6	Y	N

12 6 N N
 13 7 Y Y
 14 7 N Y
 15 7 N Y

Current View
No Of Records

Sample.mdb : viewRely
56 *Date*

12/9/1999 5:05:02 PM

RELY by CASE Match variables: ID

Matched Analysis of Tables with Non-Zero Marginals
 Matched Sets: 12 Observations: 48

Cases: 1 Controls: 3

	Exposed Controls			
Exposed Cases	3	2	1	0
1	1	1	5	4
0	0	1	1	1

SUMMARY INFORMATION

Parameters	Point	95% Confidence Interval		Upper
	Estimate	Lower		
Odds Ratio Estimates				
Crude OR (cross product)	13.0000	2.4125,	70.0530	(T)
Crude OR (MLE)	12.2200	2.4762,	94.3800	(M)
		2.0935,	133.9400	(F)
Adjusted OR (MH)	7.667	1.8621,	31.5650	(R)
Adjusted OR (MLE)	8.3589	1.9281,	58.2540	(M)
		1.6672,	81.8000	(F)
Risk Ratios (RR)				
Crude Risk Ratio (RR)	3.0000	1.6724,	5.3815	
Adjusted RR (MH)	3.0000	1.5817,	5.6902	

(T=Taylor series; R=RGB; M=Exact mid-P; F=Fisher exact)

STATISTICAL TESTS (overall assoc)	Chi-square	1-tailed p	2-tailed p
MH Chi square - uncorrected	9.5238		0.0020
MH Chi square - corrected	7.7143		0.0055
Mid-p exact		0.0014	
Fisher exact		0.0026	

In this dataset, each case had three controls. In studies with varying numbers of controls per case, several tables are produced to reflect the combinations of exposure and case status in the groups.

The interpretation of the output would be that cases were many times more likely to have the exposure (i.e., use Rely tampons) than controls, and that this is statistically significant. Because the disease is rare, the odds ratio is a good estimate of the risk ratio. One could therefore say that the risk of TSS among users of Rely tampons is 8.36 times greater than the risk among those who do not use Rely tampons, and that we are 95% confident that the true risk ratio is captured between 1.93 and 58.25 (using exact mid-p 95% confidence limits). If we did an *incorrect* analysis of matched case-control data using the TABLES command, which is essentially an unmatched analysis, the odds ratio would have been 13.

While the conclusions would be the same--that Rely tampon users have a dramatically greater risk of TSS than those who do not use Rely tampons--the analysis that ignores the matching overstates the OR. If the odds ratio is similar in the matched and unmatched analyses, the confidence intervals may also differ between matched and unmatched analyses.

Where to go after matched case-control analysis: Conditional Logistic Regression

Conditional logistic regression employs the match variable in much the same way as the previous matched analysis. An example is given in the chapter on logistic regression.

For A Single Numeric Variable—The MEANS Command

The MEANS command is used when the variable of interest is numeric and measured on a continuous scale. A continuous variable can have decimal values (real numbers like 44.645) or integer values (44). In some ways, AGE can be considered either categorical (with 1-year categories) or continuous, but to use the MEANS command the Mean or Average value of the values must be of interest. The Mean AGE of one or more groups of people is useful information, whereas the Mean of the numeric codes for countries of the world would usually be of no interest, even though both sets of data might contain numeric values.

Let's examine the output from a request for MEANS AGE in the viewOSWEGO dataset.

Current View
No Of Records

D:\EPI2000\Sample.Mdb : viewOswego
74 Date

10/28/1999 9:31:38 AM

Means of AGE

AGE	Frequency	Percent	Cum Percent
3	1	1.4%	1.4%
7	2	2.7%	4.1%
8	2	2.7%	6.8%
9	1	1.4%	8.1%
10	1	1.4%	9.5%
11	4	5.4%	14.9%
12	1	1.4%	16.2%
13	2	2.7%	18.9%
14	1	1.4%	20.3%
15	3	4.1%	24.3%
16	1	1.4%	25.7%
-----	-----	-----	-----
18	1	1.4%	31.1%
70	1	1.4%	95.9%
72	1	1.4%	97.3%
74	1	1.4%	98.6%
77	1	1.4%	100.0%
Total	74	100.0%	100.0%

Total	Sum	Mean	Variance	Std Dev	Std Err
	74 2744	37.0811	461.0344	21.4717	2.4960

T statistic=14.8560 df=73 p-value=0.0000

Comparing a Numeric Variable Across Groups—The MEANS Command with a Second Variable

The MEANS command can compare Mean values of a variable between groups of records. The numeric variable of interest, AGE, for example, is processed as for a single variable, but another categorical or “group” variable (such as ill/not ill) is used to divide the records into groups for comparison. MEANS AGE ILL, for example, compares the ages of the ill and well persons and provides statistics to evaluate whether or not there is really a difference.

If there are only two groups, the equivalent of an independent t-test is performed. If there are more than two groups, then a one-way analysis of variance (ANOVA) is computed. “One way” means that there is only one grouping variable (in the above example: ill/not ill). If there were two grouping variables, such as ill/not ill and sex, then that would be a two-way ANOVA, which Epi Info does *not* perform. The one-way ANOVA can be thought of as an extension of the independent t-test to more than two groups. Because the ANOVA test requires some assumptions about the data and the underlying population, another test (Kruskal-Wallis, also known as the Mann Whitney/Wilcoxon test if there are only two groups) is also provided. This is a non-parametric test, meaning that it does not require assumptions about the underlying population. Non-parametric tests are more conservative in detecting a statistically significant difference, but a result that is “significant” in the non-parametric test will also be so in the ANOVA test.

With a grouping variable, the MEANS command has the form:

```
MEANS <continuous var> <grouping var>
```

The output is provided in 5 different sections:

1. A table of the two variables with the continuous variable forming the rows and the grouping variable forming the columns.
2. Descriptive information of the continuous variable by each group: number of observations, mean, variance, and standard deviation; minimum and maximum values; the 25th, 50th (median), and 75th percentiles; and the mode.
3. An Analysis of Variance (ANOVA) table and a p-value for whether or not the means are equal.
4. A test to determine whether the variances in each group are similar (Bartlett's test for homogeneity of variance).
5. A non-parametric equivalent to the independent t-test and one-way ANOVA.

Using the Oswego dataset, let's examine the ages of ill and well persons in the outbreak.

MEANS AGE ILL

ILL			
AGE	(+)	(-)	TOTAL
3	1	0	1
7	1	1	2
8	2	0	2
etc	etc	etc	etc
72	1	0	1
74	1	0	1
77	1	0	1
TOTAL	46	29	75

Descriptive Statistics for Each Value of Crosstab Variable

	Obs	Total	Mean	Variance	Std Dev	
(+)	46	1806.0000	39.2609	477.2638	21.8464	
(-)	29	955.0000	32.9310	423.7094	20.5842	
	Minimum	25%	Median	75%	Maximum	Mode
(+)	3.0000	17.0000	38.5000	59.0000	77.0000	15.0000
(-)	7.0000	14.0000	35.0000	50.0000	69.0000	11.0000

ANOVA, a Parametric Test for Inequality of Population Means

(For normally distributed data only)

Variation	SS	df	MS	F statistic
Between	712.6550	1	712.6550	1.5604
Within	33340.7316	73	456.7224	
Total	34053.3867	74		

P-value = 0.2156

Bartlett's Test for Inequality of Population Variances

Bartlett's chi square= 0.1193 df=1 P value=0.7298

A small p-value (e.g., less than 0.05) suggests that the variances are not homogeneous and that the ANOVA may not be appropriate.

Mann-Whitney/Wilcoxon Two-Sample Test (Kruskal-Wallis test for two groups)

Kruskal-Wallis H (equivalent to Chi square) = 1.1612

Degrees of freedom = 1

P value = 0.2812

Total: The total is equal to $\sum f_i$.

Sum: The sum of all values is $\sum f_i x_i$, in this example 344.

Mean: The mean \bar{x}_g or average value of the observations taking into account the frequency of the values. In the example, the mean age of the observations is 4.047 years and can be calculated as the Sum/Total:

$$\bar{x}_g = \frac{\sum_{i=1}^n f_i x_i}{\sum_{i=1}^n f_i} \quad (\text{Eq 6})$$

Measures of Variability: The three measures of variability around the group mean provided are the sample group **Variance:**

$$S_g^2 = \frac{\sum_{i=1}^n f_i (x_i - \bar{x})^2}{\left(\sum_{i=1}^n f_i\right) - 1} \quad (\text{Eq 7})$$

the sample group standard deviation (**Std Dev**):

$$s_g = \sqrt{s_g^2}$$

(Eq 8)

and the standard error of the group mean (SEM) or, more simply, the standard error (**Std Err**)

$$SEM_g = \frac{s_g}{\sqrt{\sum_{i=1}^n f_i}}$$

Minimum and **Maximum** values: The smallest (minimum) and largest (maximum) values in the table.

Percentile values: The twenty-fifth (**25%ile**), fiftieth (**Median**) and seventy-fifth (**75%ile**) percentile values are provided. For the median value, if the number of observations (the TOTAL value) is an odd number, the median is calculated as:

$$\frac{n+1}{2}$$

For example, if there were 7 observations in the dataset, the median would be the value of the fourth observation in a sorted file ($[7 + 1] / 2 = 4$). If the number of observations is even, the median is calculated as the average of:

$$\frac{n}{2} \text{ and } \frac{n}{2} + 1$$

For example, if there were 8 observations in the dataset, the median would be the average of the fourth and fifth observations in a sorted file. When the mean and median differ substantially from one another, this suggests that the data are skewed. Epi Info does not currently provide a test to determine whether data are normally distributed. Calculating other percentiles is done with the following methods:

When $\frac{np}{100}$ is an integer, the percentile is the average of $\frac{np}{100}$ and $\frac{np+1}{100}$,

where n is the number of observations and p is the percentile of interest (e.g., the 25th percentile would have a p of 25). For example, if there are 20 observations in the dataset and the 25th percentile is wanted,

$(20 \times 25) / 100 = 5$. Because 5 is an integer, the 25th percentile would be the average of the values for the 5th and 6th observations in a sorted file. If $\frac{np}{100}$ is not an integer, round down to the nearest integer and add one to determine the observation for the percentile. For example, if there are 19 observations in the dataset and the 25th percentile is wanted, $(19 \times 25) / 100 = 4.75$. Round down to the nearest integer (4 in this example), then add one (for 5 in this example). Therefore, the value of the 5th observation in a sorted file would be the 25th percentile.

In the example, the 25th percentile value for cases is 17 years of age and the 75th percentile value is 59 years of age. Half of all individuals in the case group are between 17 and 59 years of age.

Mode: The most frequent observation in the table. Note that if there is more than one mode in the data, Epi Info presents *only* the smallest modal value.

The Analysis of Variance (ANOVA) table is presented next.

The calculations for the ANOVA table are as follows:

Between sums of squares (SS)

$$\sum_{i=1}^k \sum_{j=1}^{n_i} (\bar{y}_i - \bar{\bar{y}})^2$$

where

i = an observation number ranging from 1 through k (the number of groups)

j = a group number ranging from 1 through n_i (the number of observations in the i th group)

\bar{y}_i = the mean in each i th group

$\bar{\bar{y}}$ = the grand mean

or, as it is actually calculated in the software, as:

$$\sum_{i=1}^k n_i \bar{y}_i^2 - \frac{\left(\sum_{i=1}^k n_i \bar{y}_i \right)^2}{n}$$

where

n = the total sample size

The first equation for the Between SS shows that it is the sum of the differences between each group mean and the grand mean squared. The second equation is the short computational form that relies on only needing to know the sample size and mean in each group and the overall sample size.

For the Within SS

$$\sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2$$

where

y_{ij} = each observation in the dataset

or, as the Within SS is actually calculated in the software,

$$\sum_{i=1}^k (n_i - 1)s_i^2$$

where

s_i^2 = the variance in each group

The first equation for the Within SS shows how it is the sum of each observation minus the group mean squared. The second is a short computational form that requires only the sample size and variance for each group.

The Total SS is calculated by adding the Between SS and Within SS.

The Between degrees of freedom (df) is the number of groups minus 1 ($k - 1$); the Within degrees of freedom is the number of observations minus the number of groups ($n - k$).

The mean square (MS) is calculated by dividing the sums of squares (SS) by the df for both the Between and Within sources of variation. The F statistic is calculated by dividing the Between MS by the Within MS . The p-value is determined from the F statistic with $k-1$, $n-k$ degrees of freedom.

In the table that is displayed, there are six headings: variation, SS, df, MS, F statistic, p-value, and (if there are only two groups), t-value. Under the "Variation" column is the text "between," "within," and "total." When performing a one-way ANOVA, there are two sources of variability: variability between group means and variability within each group.

Most attention is usually paid to the p-value. If the p-value is <0.05 (or some other value used to define statistical significance), then there is a statistically significant difference between two or more of the means. The ratio of the “Between MS” (Mean Square) and “Within MS” is what makes the *F statistic*, in this example, 1.560. If the p-value is ≥ 0.05 , then there is no statistically significant difference between the means. In the ANOVA, the *F statistic* is always provided. If there are only two groups, the *t-value* is also provided; this same t-value would be derived through an independent t-test. The t-value is the square root of the F statistic, so whichever is used (F or t), the p-value is the same. (Note that in this discussion the value “0.05” was used to define “statistical significance.” This was only done for illustrative purposes and some investigators may choose a different value, such as 0.10 or 0.01.)

To compute confidence intervals around the mean values, use the following formula:

$$\text{Mean} \pm (\text{t-value}) * (\text{Within MS}/n_j)^{0.5}$$

The above formula is similar to the one discussed for the FREQ command, but there are differences. Instead of the standard error, there is $(\text{Within MS}/n_j)^{0.5}$, that is, the square root of the within mean square is divided by the number of individuals in the group. The use of a standard error based on the within mean square is more accurate than if the variance for each group was used separately. In the example, the variance in age for those who are ill is 477 and for those not ill, 420; note that the within mean square is 456. Also note that the Bartlett’s test for homogeneity of variance in the above example found that the variances can be assumed to be the same.

```
Bartlett's test for homogeneity of variance
Bartlett's chi square = 0.119 deg freedom = 1
p-value = 0.729786
```

One of the assumptions of the ANOVA is that the variances in each of the groups are similar. One way to test this assumption is through use of Bartlett’s test for homogeneity of variance, a fancy way of asking, “Are the variances about the same?” The variances are compared and a chi square value, the degrees of freedom, and p-value are presented. A note is also provided on the screen about how to interpret the results. In the above example, because the p-value is ≥ 0.05 , the variances can be assumed to be equal. If the variances are not similar, then: 1) the non-parametric results discussed next should be used; and 2) the data could be transformed.

```
Mann-Whitney/Wilcoxon Two-Sample Test (Kruskal-Wallis test for 2
groups)
```

Kruskal-Wallis H (equivalent to Chi square) =	1.161
Degrees of freedom =	1
p value =	0.281226

The above result provides a non-parametric equivalent to the ANOVA. The non-parametric results should be used when: 1) the means of each group is not a good measure of centrality (i.e., the data are skewed or have some other non-normal distribution); 2) the data are rankings or ordinal data rather than precise numeric values (e.g., using Likert scales where 1=strongly agree and 7=strongly disagree with intermediate levels of agreement is an example of ordinal data; compare this to cholesterol values, which are precise quantitative values); 3) there is a small sample size in each group with numeric data (which Rosner suggests should be less than 10 in each group); and 4) the variances are significantly different between the groups. (NOTE: The t-test for independent samples with unequal variances is a procedure with two groups that can be used when the variances are not equal, *t*, but this test is not available in Epi Info. The non-parametric equivalent to the t-test where two groups are compared is the Mann-Whitney U test, the Wilcoxon rank-sum test (called the Wilcoxon two-sample test in the output from *Analysis*). If there are more than two groups being compared, the Kruskal-Wallis test is the non-parametric equivalent to ANOVA.

In the example above, the p-value is 0.281226, which would indicate that there is little difference in the rankings of age between those who were ill and those who were not ill, leading to the same conclusion as the p-value from the ANOVA table.

Evaluating Quantitative Differences (e.g., Before and After) in the Same Records—MEANS with a Variable Representing a Difference

Sometimes two numbers in the same record represent paired numeric values. This happens in evaluations where examinations are given before and after an educational program or measurements of a quantitative variable are taken before and after an intervention. Did the students do better on the same examination after taking a course? Did heart rate or blood pressure change in response to a medication?

Of course, separate records could be created for each instance of the test and another field could be used to indicate whether the test is “before” or “after.” The MEANS command could then compare mean group values before and after the event.

This method, however, ignores the matching that occurs from using the same student or subject for both tests. A better method is to subtract the “before” score from the “after” score to find the difference for each student, and then to see if the average difference is significantly different from zero, using Student’s t-test. Epi Info performs the t-test every time the MEANS command is given with a single variable, just in case the variable represents a difference and you would like to know if it differs, on the average, from zero.

Student's "t": The statistic presented in the MEANS (one variable) command is for comparing the mean of the observations with a null value of zero. Although this statistic may be useful in some circumstances, frequently it is not. In the above example, determining whether the mean age differs from zero does not make much sense. However, it would make sense in the following example. Using an example from Rosner (1995, see Table 8.1), systolic blood pressure measurements are obtained from 10 women; one measurement from each woman is performed at baseline, prior to oral contraceptive usage, and then a second measurement is taken while the women are using oral contraceptives. If oral contraceptives do not affect blood pressure, then we would expect that the difference between measurements on each woman should, on average, be zero. If oral contraceptives *do* affect blood pressure, then the second measurement should differ from the first. The data are shown in Table 3.

Table 3. Example of paired data (from Rosner, 1995)

<u>Women No.</u>	<u>Measure 1</u>	<u>Measure 2</u>
1	115	128
2	112	115
3	107	106
4	119	128
5	115	122
6	138	145
7	126	132
8	105	109
9	104	102
10	115	117

Create a project file, and name it Practice.mdb. Then make a view called viewBPDifference and include the following variables: Woman No., Measure 1, and Measure 2. Enter the data, and **Analysis** will calculate the difference with the following program.

Current View: C:\EPI2000\Practice.MDB : viewBPDifference
 Record Count: 10 Date: 10/22/1999 7:39:46 AM

```
Read 'C:\EPI2000\Practice.mdb': viewBPDifference
Define Diff
Assign Diff = Measure2 - Measure1
Freq Diff
Means Diff
```

The results are:

Frequency command for a paired t-test

Current View: C:\EPI2000\Practice.MDB : viewBPDifference
 Record Count: 10 Date: 10/22/1999 7:39:46 AM

Frequency of DIFFERENCE

Difference	Frequency	Percent	Cumulative Percent	
-2	1	10.0%	10.0%	
-1	1	10.0%	20.0%	
2	1	10.0%	30.0%	
3	1	10.0%	40.0%	
4	1	10.0%	50.0%	
6	1	10.0%	60.0%	
7	2	20.0%	80.0%	
9	1	10.0%	90.0%	
13	1	10.0%	100.0%	
Total	10	100.0%	100.0%	

95% Confidence Limits

-2	0.3%	44.5%
-1	0.3%	44.5%
2	0.3%	44.5%
3	0.3%	44.5%
4	0.3%	44.5%
6	0.3%	44.5%

Epi Info 2000

Statistics

7 2.5% 55.6%
 9 0.3% 44.5%
 13 0.3% 44.5%

Current View: C:\EPI2000\Practice.MDB : viewBPDifference
 Record Count: 10 Date: 10/22/1999 7:39:46 AM

Means of DIFFERENCE

Difference	Frequency	Percent	Cum Percent
-2	1	10.0%	10.0%
-1	1	10.0%	20.0%
2	1	10.0%	30.0%
3	1	10.0%	40.0%
4	1	10.0%	50.0%
6	1	10.0%	60.0%
7	2	20.0%	80.0%
9	1	10.0%	90.0%
13	1	10.0%	100.0%
Total	10	100.0%	100.0%

Total	Sum	Mean	Variance	Std Dev	Std Err
10	48	4.8000	20.8444	4.5656	1.4438

Student's "t", testing whether mean differs from zero.
 T statistic=3.3247 df=9 p-value=0.0044

The Student's t-test statistics make sense in this example and show that, on the average, a woman's systolic blood pressure will increase after the use of oral contraceptives by 4.8 mm Hg, and that this increase is significantly different from zero (p-value = 0.00887). This t-statistic is calculated by:

$$t = \frac{\bar{x}}{SEM}$$

A 95% two-sided confidence interval around the mean difference can be calculated. In this example, the t-value for a 95% confidence interval would need to be obtained from a textbook. For a large number of observations ($n \geq 120$), the value of 1.96 can be used. In the above example, look up the t-value for $n-1$ degrees of freedom ($10 - 1 = 9$) for the value for a two-sided 95% confidence interval, which is 2.262. Applying this to the results above gives $4.8 \pm 2.262 * 1.444 = (1.53, 8.07)$. The interpretation is that we are 95% confident that the true increase in systolic blood pressure is captured between the values of 1.5 mm Hg and 8.1 mm Hg.

REGRESS Command

REGRESS can be used for simple linear regression (only one independent variable), for multiple linear regression (more than one independent variable), and for quantifying the relationship between two continuous variables (correlation). Regression is used when the primary interest is to predict one dependent variable (y) from one or more independent variables (x_1, \dots, x_k).

The correlation coefficient or r (sometimes referred to as the Pearson correlation coefficient) is a useful measure of how two continuous variables are related. If the correlation is greater than 0, the variables are **positively correlated**; as x increases, y also increases. If the correlation is less than 0, the variables are **negatively correlated**; as x increases, y decreases. If the correlation is exactly 0, then the variables are **uncorrelated**. The correlation coefficient can vary between +1 and -1. For positive correlations ($r > 0$), the closer to +1 the stronger the correlation; for negative correlations ($r < 0$), the closer to -1 the stronger the correlation.

If the data are ordinal or far from normal, significance tests based on the Pearson correlation coefficient are not valid and a non-parametric equivalent to Pearson's should be used.

Simple Linear Regression

Using the data from Rosner (1995) again, here is a simple linear regression on the relation between Estriol and Birthweight (BW):

Current View: D:\EPI2000\Sample.Mdb:viewEstriolAndBirthweight

Record Count: 31 *Deleted Records:* Excluded *Date:* 1/15/00 11:17:14 AM

REGRESS BIRTHWEIGHT = ESTRIOL

Birthweight = Estriol

Correlation Coefficient: $r^2=0.37$

Source	df	Sum of Squares	Mean Square	F-statistic
Regression	1	248.421	248.421	16.811
Residuals	29	428.547	14.777	
Total	30	676.968		

Variable	Coefficient	Std Error	F-test	P-Value
Intercept	21.536	2.636	66.7390	0.000000
Estriol	0.606	0.148	16.8108	0.000076

Correlation coefficient The Pearson correlation coefficient, or “r”. In this example, the correlation is 0.61, indicating a relatively strong positive correlation between estriol and birthweight. With only a single independent variable, the correlation = square root (R^2).

r^2 Sometimes represented as r^2 or R^2 , i.e., R squared. The R^2 value = Regression Sum of Squares / Total Sums of Squares (in the above example, $250.5745/674 = 0.37177$). The R^2 can be thought of as the proportion of variance of y (in this example, birthweight) that can be explained by x (in this example, estriol). In this example, 37% of the variance in birthweight can be explained by the women’s estriol levels. If $R^2 = 1$, then all of the variability is explained, which would mean that all data points fall on the regression line. If $R^2 = 0$, then no variance is explained. A 95% confidence interval for the R^2 value is also provided (0.02, 0.64). (Note: The confidence interval type can be changed, for example to 90%, with the command SET CONFIDENCE= 90.)

F-Statistic The F-statistic is the Regression Mean Square / Residual Mean Square (in the above example, $250.5745/14.6009 = 17.16$). The F-statistic is calculated to determine if the slope of the regression line is significantly different from 0. Epi Info does not provide the p-value.

Mean The average value for ESTRIOLE; could also be determined with `FREQ ESTRIOLE`.

Coefficient The slope in the line, sometimes referred to as the “regression coefficient.” In this example, 0.608 can be interpreted as: For every 1 unit increase in estriol (1 mg/24 hr), there is a 0.61 increase in each birthweight unit (g/100). Statistics concerning the slope are also provided: the standard error (Std Error), 0.146812, and the 95% confidence interval (0.307921, 0.908460). The interpretation would be that

although we observed for every 1 unit increase in Estriol a 0.61 increase in birthweight, we are 95% confident that the “true” slope would be captured between 0.31 and 0.91. (As above, the confidence interval type can be changed, for example to 90%, with the command SET CONFIDENCE= 90.)

Y-Intercept This is where the line intercepts the y line. In this example, the line would intercept the (birthweight) line (y) at 21.5.

The general form of the simple linear regression line is:

$$y = a + bx$$

where y is the dependent variable, a is the intercept, and x is the independent variable.

In the above example, the regression line is:

$$\begin{aligned} BW &= a + b (\text{estriol}) \\ BW &= 21.52 + 0.608 (\text{estriol}) \end{aligned}$$

For any given value of estriol, a BW value can be predicted. For example, using the mean estriol level of 17.226,

$$BW = 21.523 + 0.608(17.226) = 31.996$$

Within Epi Info, the predicted values and the residuals could be added to the file. In the above example:

```
DEFINE PREDICTED ##.#####
LET PREDICTED = 21.523 + 0.608 * ESTRIOL
DEFINE RESIDUAL ##.#####
LET RESIDUAL = BW - PREDICTED
```

Multiple Linear Regression

Using data from Rosner (1995, Example 11.2.4, page 470), the dependent variable is systolic blood pressure, and the independent variables are birthweight in ounces and age in days.

Current View: **D:\EPI2000\Sample.Mdb:viewBabyBloodPressure**

Record Count: **16 Deleted Records: Excluded Date: 1/15/00 11:20:48 AM**

REGRESS SystolicBlood = AgeInDays Birthweight

SystolicBlood = AgeInDays Birthweight

Correlation Coefficient: $r^2=0.88$

Source	df	Sum of Squares	Mean Square	F-statistic
Regression	2	591.036	295.518	48.081
Residuals	13	79.902	6.146	
Total	15	670.938		

Variable	Coefficient	Std Error	F-test	P-Value
Intercept	53.450	4.532	139.1042	0.000000
AgeInDays	5.888	0.680	74.9229	0.000000
Birthweight	0.126	0.034	13.3770	0.000724

ra² This is an adjusted r^2 value that takes into account the model's associated degrees of freedom. The ra^2 value = $1 - [\text{Total df}/\text{Residual df}] * [\text{Residual SS} / \text{Total SS}]$ (in the above example, $1 - [15/13] * [79.9019/670.9375] = 0.86$). The ra^2 can be thought of as an r^2 value that adjusts for the number of independent variables. Unlike the r^2 , which will always increase as the number of independent variables increase, the ra^2 can become smaller.

The F-statistic can be used to determine the p-value by use of a table. Find the p-value of F (numerator df), (denominator df), (F-statistic), which in this case is F,2,13,48.08, in which the p-value is < 0.001 , thus concluding that the two variables together are significant predictors of SBP.

The regression line is:

$$SBP = 53.4501940 + 0.1255833*BWT + 5.8877191*AGE$$

To predict the SBP in an infant who weighs 128 oz and 3 days old is (rounding to 3 decimal places):

$$SBP = 53.450 + 0.126*(128) + 5.888*(3)$$

SBP = 87.242 mm Hg

Sample Size Calculations in *Statcalc* for DOS

The sample size calculations for proportions in a descriptive study or survey use the following method:

Sample size = size of sample randomly selected from the population

Population = size of population which the sample is to represent

P = true proportion of factor in the population (guess)

D = (Maximum) difference between sample mean and population mean

Z = area under normal curve corresponding to the desired confidence level.

<u>Confidence=</u>	<u>Z</u>
.90	1.645
.95	1.960
.99	2.575
.999	3.29

The formula is:

$$\text{Sample size} = n / (1 + (n / \text{population}))$$

$$\text{in which: } n = Z^2 * P(1-P) / D^2$$

Reference: Kish, 1965.

The sample size formula for comparison of two groups of equal or unequal size is taken from Fleiss (1), pp. 44-45, formulas 3.19 and 3.20, and table A.2 on p. 259. These formulas do not assume matching of the two populations, and they apply to univariate comparisons of the differences in proportions (rates) in the two populations.

m = size of sample from population 1

r*m = size of sample from population 2

P1 = true proportion of factor in population 1

P2 = true proportion of factor in population 2

alpha = chance of falsely declaring the two proportions to differ ("Significance")

beta = chance of not detecting a difference which is present
(could be called "Lack of sensitivity")

1-beta = chance of detecting a real difference ("Power").

In the equations the values of c are as follows:

If 1-alpha is:	Then c[alpha/2] is:
.90	1.645
.95	1.960
.99	2.575
.999	3.29

If 1-beta is:	Then c[1-beta] is:
.70	-0.524
.80	-0.842
.90	-1.282
.95	-1.645
.99	-2.326

These correspond to Table A.2 in Fleiss (1981), but it takes considerable attention to detail and to the example on p. 45 to learn how to look them up correctly.

The formulas are:

$$m = m_{\text{prime}} + (r+1)/(r|P_2 - P_1|)$$

in which "|" denotes absolute value and:

$$M_{\text{PRIME}} = (((C_{[\text{ALPHA}/2]} * \text{SQRT1}) - (C_{[1-\text{BETA}]} * \text{SQRT2}))^2) / (R * ((P_2 - P_1)^2))$$

in which "[]" denotes a subscript and:

$$\text{SQRT1} = ((R+1) * P_B * Q_B)^{0.5}$$

$$\text{SQRT2} = ((R * P_{R1} * (1 - P_{R1})) + (P_{R2} * (1 - P_{R2})))^{0.5}$$

$$P_B = (P_{R1} + (R * P_{R2})) / (R+1)$$

$$Q_B = 1 - P_B$$

Acknowledgements

We wish to thank Drs. David Martin and Harland Austin for use of their source code for computing exact and mid-p exact statistical tests and confidence intervals for the odds ratio and for the rate ratio. Thanks to those who reviewed the this chapter and provided comments, and to those who tested the software.

References

- Kish L. Survey sampling. New York: John Wiley & Sons; 1965.
- Fleiss JL. Statistical methods for rates and proportions. 2nd Edition. New York: John Wiley; 1981.
- Greenberg RS, Daniels SR, Flanders WD, Eley JW, Boring JR. Medical epidemiology. Norwalk, Connecticut: Appleton & Lange; 1996.
- Hansen MH, Hurwitz WN, Madow WG. Sample survey methods and theory. Vol. I. New York: John Wiley; 1953.
- Kahn HA, Sempos CT. Statistical methods in epidemiology. New York: Oxford University Press; 1989.
- Kelsey JL, Whittemore AS, Evans AS, Thompson, WD. Methods in observational epidemiology. 2nd Edition. New York: Oxford University Press; 1996.
- Kleinbaum DG, Kupper LL, Morgenstern H. Epidemiologic research: Principles and quantitative methods. Belmont (CA):Lifetime Learning Publications; 1982.
- Martin D, Austin H. An efficient program for computing conditional maximum likelihood estimates and exact confidence limits for a common odds ratio. *Epidemiology* 1991; 2(5):359-362.
- Rosner B. Fundamentals of biostatistics. 4th Ed. New York: Duxbury Press; 1995.
- Rothman KJ. Modern epidemiology. Boston: Little, Brown; 1986.
- Schlesselman JJ. Case-control studies. New York: Oxford University Press, 1982.
- Wolter KM. Introduction to variance estimation. New York: Springer-Verlag; 1985.
- Woodruff RS. A simple method for approximating the variance of a complex estimate. *Journal of the American Statistical Association* 1971; 66:411-414.

Logistic Regression – The MVAWin Program



Questions and Answers

Cecile Delhumeau and Andrew Dean

How Do I Know When I Need Logistic Regression?

In Epi Info 2000, either the [TABLES](#) command or logistic regression (LOGISTIC command) can be used when the outcome variable is dichotomous (for example, disease/no disease). Analysis with the TABLES command in Epi Info is adequate if there is only one “risk factor.” Logistic regression is needed when the number of explanatory variables (“risk factors”) is more than one. The method is often called “multivariate logistic regression.”

Logistic regression shows the relationship between an outcome variable with two values and explanatory variables that can be categorical or continuous.

What Kinds of Epidemiologic Data Sets Are Particularly Suitable for Logistic Regression? Not Suitable?

Logistic regression can be applied to case-control, follow-up, and cross-sectional data. All types of variables (categorical and continuous) can be included in a logistic regression model, although categorical (coded) variables make results easier to interpret. The outcome variable must be dichotomous. Categorical data must be coded in "0/1", "0" being the class with the least exposure (Epi Info takes care of this for YES/NO variables).

Types of Variables

Types of variable	Other terms ...	Examples ...
Continuous variables	Quantitative variables	Weight, Blood pressure
Categorical variables	Qualitative variables, discrete variables, nominal variables,	Presence or absence of a disease or risk factor, Race

coded variables.

What is a Logistic Regression “Model”?

The purpose of logistic regression is to produce a mathematical equation that relates the probability of an outcome to the particular value of risk factor variables. A model might predict the probability of occurrence of a myocardial infarction (MI) over a 5-year period, given a patient’s age, sex, race, blood pressure, cholesterol level, and smoking status.

In Epi Info, a model can be expressed as:

MI = age + sex + race + blood pressure + cholesterol + smoking status

The actual model is

$$MI = 1 + e^{-(\alpha + \beta_1 * age + \beta_2 * sex + \beta_3 * race + \beta_4 * blood_pressure + \beta_5 * cholesterol + \beta_6 * smoking_status)}$$

or

$$\text{logit_MI} = \alpha + \beta_1 * age + \beta_2 * sex + \beta_3 * race + \beta_4 * blood_pressure + \beta_5 * cholesterol + \beta_6 * smoking_status$$

Epi Info 2000 uses a process called “maximum likelihood estimation” to arrive at the best estimate of the relationships based (usually) on a follow-up study, such as the well known Framingham study. The results include values for the beta coefficients (“ β ”) but more important for epidemiologists, can produce an odds ratio (OR) for each value of a risk factor compared with its baseline (“absent” or “normal”) state.

How Do I Pick the Outcome Variable?

The outcome variable is a dichotomous variable (two values) for which other variables may provide an explanation. Often, the outcome variable will indicate the presence or absence of a disease. To explore the risk factors for premature birth, however, the outcome variable might be low birthweight.

The coding of this variable must be in "0/1":

"1" for persons who experienced the event studied (disease or low weight),

**Epi Info 2000
Program**

Logistic Regression – The MVAWin

"0" for persons who did not (no disease or normal weight).

NOTE: Epi Info does the coding if the outcome variable is a “yes / no” variable.

How Can I Use Explanatory Categorical Variables with More Than 2 Values?

Explanatory categorical (coded) variables with more than 2 values must be recoded with dummy variables. If not, these variables will be considered to be quantitative variables, which may make it difficult to interpret the results of the logistic regression model. A variable with "k" values must be coded in "k -1" dichotomous dummy variables that have two values, with "0" value indicating no or least exposure.

For example:

	coding	
Race: White		1
Black		2
Asian/Pacific Islander		3

Dichotomous dummy variables coding used for Race:

	Design	Variables
<i>Race</i>	R1	R2
White	0	0
Black	1	0
Asian/ Pacific Islander	0	1

Two dichotomous dummy variables are enough to locate the three initial values of the "Race" variable. In the logistic regression model, only the variables R1 (Black Race = 1) and R2 (Asian/Pacific Islander = 1) will appear, and the risk computed will be in reference to White Race (risk = 1).

NOTE: Epi Info creates dummy variables for categorical variables.

How Can I Use Explanatory Continuous Variables?

If there are continuous variables in the dataset (age, or duration of a treatment, for example), it is

possible to include them as continuous variables in the logistic regression model. The results are expressed as the risk for each unit of the continuous variable (each year of age or each day of a treatment, for example) and the risk for the number of units desired will have to be computed (see examples below, under "What Does the Model Look Like in Epi Info?").

Continuous variables can also be recoded into categories to make interpretation easier, using the median as a cut point to recode them in two values, or quartiles as a cut point to recode them in four values, or other meaningful categories (e.g., age = adult / child or 10 year age groups, treatment = none, partial, or full). After recoding, test each version in a model containing only outcome and the variable being recoded. Choose the coded or continuous version that gives the most significant result in the likelihood ratio test (the smallest).

NOTE: The interpretation of the results for a continuous variable is more difficult than for a categorical variable, but in recoding a continuous variable in a categorical variable, some information is lost from the data. It is never a simple choice.

What is Interaction and Why Should It Be Addressed Early in the Model?

Interaction means that the odds ratio (OR) for a variable varies with the value of another variable. For example, if the outcome is mesothelioma, a form of lung cancer caused by asbestos exposure, the effect (OR) of asbestos exposure differs greatly between smokers and nonsmokers.

Mesothelioma = asbestos + smoking

does not tell the whole story, and another term, called an "interaction term," is needed:

Mesothelioma = asbestos + smoking +asbestos*smoking

Interaction must be addressed early in forming a model, because the model must contain all single variables that are found in significant interaction terms to be "hierarchically well structured." One school of thought in modeling strategy suggests that all pertinent interaction terms be evaluated before eliminating any individual variables.

After Choosing the Outcome Variable, How Do I Construct a Good Logistic Regression Model?

"Different people have different modeling strategies. It's important that you develop a point of view for yourself (after considering other schools of thought)." D.G. Kleinbaum

We give you one strategy below. The references contain other modeling strategies to consider (Hosmer, Breslow).

One modeling strategy to find risk factors for an outcome involves two stages:

- A. Variable specification and interaction assessment;
- B. Confounding assessment followed by consideration of precision.

A. Variable and interaction specification: Choose the variables to include in the multivariate logistic regression model.

If you have only a few variables, start with all of them. Otherwise, include variables that may be risk factors or control variables, based on literature review, and then add all variables for which the p-value of the chi square, Fisher exact, or t-test is less than 0.25 (not the usual threshold of 0.05) in the Epi Info [TABLES](#) or [MEANS](#) commands. (Note: If you have several exposure and control variables, you have to build a model with all of them and their pertinent interactions.)

Now that you have an outcome variable and a list of confounder/risk variables, choose one variable as your primary effect variable for analysis. In the L.A. Study example, the outcome variable is Ill (yes/no). To examine the role of *estrogens*, choose it as the effect variable.

Construct the model as follows:

Ill = estrogen + age + hypertension + age*hypertension + estrogen*age + hypertension*estrogen

NOTE: If there are a lot of exposure variables (for example, in the Oswego study there are 20 variables), build a first model with all of the variables, choose those variables with the smallest p-value (around 7) and build another model with these variables and their pertinent interactions (biologically meaningful subsets).

Use a *backward elimination approach* to find the best model. This means eliminating variables or groups of variables one at a time, keeping only those that are “meaningful” in the model. “Meaningful” in this case stands for a p-value < 0.05 and the likelihood ratio test < 0.05 for the model containing this “chunk” versus the one from which it has been removed.

Frequently there is “interaction” among variables so that the OR for one depends on the value of another (e.g., the effect of age or smoking might vary in different races).

If you have two or more categorical explanatory variables in the best model, you must systematically test the interaction term in the logistic regression model (see previous question). The interaction term in the logistic regression model corresponds to the mathematical product of the two variables (variable 1*variable 2). In this case, interaction variables are included in the model, as smoking*race and ageclass*race. If you have more than two categorical variables included in the model, you must test the interaction term for biologically meaningful subsets (which may be all of them, depending on the literature). If there is an interaction term, it is important information for the study: the relationship between the exposure variable and the outcome variable is not the same for all levels of the explanatory variable (the risk factor varies).

All variables in interaction terms must be included elsewhere in the model as single terms, or Epi Info will not allow the interaction term. This results in what is known as a “hierarchically well formulated” model. Once an interaction has been found to be significant, all of its smaller subcomponents must be left in subsequent models.

The simplest interaction terms are pairs of categorical variables. In setting up the model, list the single variables first. The following example is from the Oswego study:

vanilla, chocolate, age

Then include the interaction term:

age + vanilla + chocolate + vanilla*chocolate + vanilla*age + chocolate*age (interaction term)

Third-order terms should also be included, as in vanilla*chocolate*age, if pertinent.

Use the likelihood ratio (“chunk”) test to assess the effect of removing the third-order interaction term at the same time. Run the full model, then remove the three-variable term and run it again. The results of the log likelihood ratio test for the “chunk” removed will appear at the bottom of the output. If the likelihood ratio p-value is < 0.05, at least part of the “chunk” is significant. Now do the same for the second-order terms, removing both at one time. If the likelihood ratio test p-value is significant, then at least one of the two terms is significant, and they should be evaluated individually. Identify any two variables and single variable terms that are components of significant three-variable terms. These must stay in the model, regardless of subsequent results. They are protected members of the “hierarchically well formulated model.”

More details on interaction can be found below, under " How Do I Interpret the Results of Logistic Regression Containing Interaction Terms?"

B. Confounding assessment.

At this point you have what is called the “Gold Standard” model, because the estimated OR for the exposure variable is likely to be the most “correct.” Confounding is assessed without the use of statistical testing. The procedure involves determining whether the estimated OR changes meaningfully in comparing the OR of the best model versus the model without one or more possible confounders. If there is no interaction, the assessment of confounding is carried out by monitoring changes in the OR of the explanatory variable. However, if there is interaction, the assessment of confounding is more subjective, because it requires comparison of the OR of the exposure variable with the significant interaction terms. The goal is to find a model that gives OR estimates for the exposure variable and interaction terms similar to those given by the “Gold Standard” model. If confounders can be removed without changing these ORs, and the precision (width of the confidence interval) of the OR for the exposure improves, this should be done. Care should be taken not to remove any confounders used in interaction terms.

In the Oswego study, we find:

Ill = bakedham + mashedpota + brownbread + milk + coffee + cakes + vanilla + relevant interaction

and the model becomes:

Ill = vanilla

I Have Seen a Lot of Arguments in the Literature About the Improper Use of Regression Analysis, Even Among Experts. What Can Cause Wrong Conclusions? How Do I Know That I Have Used It Correctly?

If you didn't take into account a confounder or an interaction factor, your conclusion may be incorrect. To avoid this, use:

- * Good modeling strategy (see above, "What is Interaction and Why Should It Be Addressed Early in the Model?"):
 - good choice of the variables included in the multivariate logistic regression model
 - good choice of the coding of variables (very important!)
 - appropriate assessment of interaction terms
 - appropriate assessment of the confounder variables.

* Quality of the results:

precision of confidence interval: CI 0,95 ($B_i \pm 1.96sB_i$)

is the value "1" inside CI 0,95 or not?

is Wald test of the differences among variables significant?

is the likelihood ratio of the chi square test of the model significant?

comparison of the results with other studies

analysis of residuals of the model.

What Does the Model Look Like in Epi Info?

There are two examples provided with Epi Info 2000:

1) **Oswego study: A study about an outbreak of acute gastrointestinal illness.**

We use this example to illustrate the fact that we don't need logistic regression for this study. On April 19, 1940, the local health officer in the village of Lycoming, Oswego County, New York, reported the occurrence of an outbreak of acute gastrointestinal illness to the district Health Officer in Syracuse. Seventy-five persons ate at a particular supper, and 46 persons with gastrointestinal illness were identified. The goal for the study was to find which food or foods caused the outbreak.

The outcome variable is *Ill* (yes/no).

Possible risk factors (predictor variables) are *foods and drinks consumed*.

This file is available in SAMPLE.MDB as viewOSWEGO.

2) **L.A. Study: Los Angeles retirement community study of endometrial cancer.**

We use this example to illustrate the methods for matched data analysis in the study of the effect of exogenous estrogens on the risk of endometrial cancer reported by Mack et al. (1976). These investigators identified 63 cases of endometrial cancer in a retirement community near Los Angeles. Each case was matched to 4 control women who were alive and had lived in the

community one year at the time the case was diagnosed, and who had entered the community at approximately the same time. In addition, controls were chosen from among women who had not had a hysterectomy prior to the time the case was diagnosed, and who were therefore still at risk for the disease. Information on the history of use of several specific types of medicines, including estrogens, antihypertensives, sedatives, and tranquilizers, was abstracted from the medical record of each case and control. Other abstracted data relate to pregnancy history, mention of certain diseases, and obesity.

The analysis of these data is aimed at studying the risk associated with the use of estrogens, as well as with a history of gallbladder disease, and how these risks may be modified by other explanatory variables.

Variables in the file include:

The outcome variable, *Ill* (yes/no)

Possible risk factors (predictor variables): *age*, *hypertension*, *gallbladder disease*, *other drugs* (non-estrogen), *estrogens* (any) *conjugated estrogen: amount* (mg/day), or *conjugated estrogen: duration* (months).

After I Have a Model, What Can I Do with It?

The use of a model depends on the design of the epidemiologic study. When using follow-up data, you can estimate a predicted risk (P), then calculate a relative risk (RR). In using case-control or cross-sectional data, however, a limitation is that you cannot estimate individual risks even though you can still obtain an OR. This limitation is not severe if the goal of the study is to obtain a valid estimate of an exposure-disease association in terms of an OR. As in univariate analysis, the OR becomes a reasonable estimate of the RR if the disease is “rare.”

Take, for example, a follow-up study:

The fitted model that we obtained by using the method of the maximum likelihood to estimate the parameters of the variables is:

$$\text{Logit}(D) = -3.911 + 0.652 * \text{cat} + 0.029 * \text{age} + 0.342\text{ecg}$$

Where;

D, the disease, is coded "0" if no disease

"1" if disease

cat stands for catecholamine level, coded "0" if low level

"1" if high level

Epi Info 2000 Program

Logistic Regression – The MVAWin

age is a continuous variable
ecg denotes electrocardiogram status, coded "0" if normal
"1" if abnormal.

The predicted risk of this model is:

$$P(D = 1 / cat, age, ecg) = \frac{1}{1 + e^{-(\log it(D))}} = P(X)$$

Suppose we want to use our fitted model, to obtain the predicted risk for a certain individual, for example:

cat = 1; age=40 and ecg=0.

$$P1(X) = \frac{1}{1 + e^{-(3.911+0.652(1)+0.029(40)+0.342(0))}} = \frac{1}{1 + e^{-(-2.101)}} = \frac{1}{1 + 8.173} = 0.109$$

For a person with cat=1, age=40, and ecg=0, the predicted risk obtained from the fitted model is 0.109. That is, this person's estimated risk is about 11% over the time period studied.

cat = 0; age=40 and ecg=0.

$$P0(X) = \frac{1}{1 + e^{-(3.911+0.652(0)+0.029(40)+0.342(0))}} = 0.060$$

For a person with cat=0, age=40, and ecg=0, the predicted risk obtained from the fitted model is 0.060. That is, this person's estimated risk is about 6%.

$$\frac{P1(X)}{P0(X)} = \frac{0.109}{0.060} = 1.82 = RR$$

Here, for the same fitted model, we compare the predicted risk of a person with cat=1, age=40, and ecg=0 with that of a person with cat=0, age=40, and ecg=0. If we divide the predicted risk of the person with high catecholamine with the predicted risk of the person with low catecholamine, we get an RR estimate of 1.82. Thus, using the fitted model, we find that the

person with high cat has almost twice the risk of the person with low cat, assuming both persons are of age 40 and have no previous ecg abnormality.

Two conditions must be specified to estimate RR directly. First, we must have a follow-up study; second, for two individuals being compared, we must specify values for all the independent variables in our fitted model to compute risk estimates for each individual.

If either of the above conditions is not satisfied, then we cannot estimate RR directly. We can only compute an OR that is a measure of the association directly estimated from a logistic regression model (without requiring special assumptions), regardless of whether the study design is follow-up, case-control, or cross-sectional. As in univariate analysis of a case-control study, the OR is a reasonable estimate of RR only if the disease is rare.

$$OR_{x1/x0} = e^{\sum \beta_i(x_{1i}-x_{0i})}$$

So, for the same example:

X1= (cat=1; age=40; ecg=0)

X0= (cat=0; age=40; ecg=0)

$$OR = e^{\beta_1(1-0)+\beta_2(40-40)+\beta_3(0-0)}$$

$$OR = e^{\beta_1+0+0}$$

$$OR = e^{\beta_1} = 1.91 = \text{adjusted OR}$$

Using the fitted model, we find that the person with high cat has almost twice the risk of the person with low cat, OR adjusted on the age and the ecg.

For each dichotomous variable, it is possible to compute an adjusted OR.

If X_i is coding in (0,1): $OR = e^{\beta_i}$

If X_i is coding in (a,b): $OR = e^{\sum (X_{\alpha_i}-X_{\beta_i})\beta_i}$

For the continuous variable, it is the same formula but you must multiply your β_i with a coefficient (e.g., for the age you can compute an adjusted OR).

If we compare the effect for a person of 40 years of age with that for a person of 35 years of age, then adjusted $OR = e^{(40-35)\beta_{age}} = 1.15$

So, a person of 40 years of age will have 1.15 times more risk of having the disease than a person of 35 years of age, OR adjusted on cat and ecg.

How Do I Interpret the Results of Logistic Regression Containing Interaction Terms?

If there is an interaction term (say $E \cdot X$ in a logistic regression model), then the risk factor E for outcome M differs with the values of the explanatory dichotomous variable X .

In mathematical expressions:

$$\text{Logit}P = \alpha + \beta_1 * E + \beta_2 * X + \gamma E * X$$

interaction term

if $X = 0$, then

$$\ln OR = \alpha + \beta_1 - \alpha = \beta_1$$

and if $X = 1$, then

$$\ln OR = \alpha + \beta_1 + \beta_2 + \gamma - (\alpha + \beta_2) = \beta_1 + \gamma$$

There is an interaction between E and X if γ is different than "0" (Wald test (Z-Statistic) or the likelihood ratio chi square test is significant on the threshold of 0.05).

For example:

$$\text{Logit}(P) = -0.39 + 0.06 _tobacco + 0.19 _inhale + 0.82 _tobacco * inhale$$

interaction term

**Epi Info 2000
Program**

Logistic Regression – The MVAWin

Coding of tobacco: no tobacco "0"
tobacco "1"

Coding of inhale: don't inhale "0"
inhale "1"

There are two ORs: one for people who inhale the tobacco (inhale = 1) and one for people who don't inhale the tobacco (inhale = 0). Two ORs are computed.

$$OR_{\text{people who don't inhale tobacco}} = e^{0.066} = 2.43$$

$$OR_{\text{people who inhale tobacco}} = e^{(0.066+0.82)} = 1.07$$

The people who inhale tobacco have 2.43 more risk to be ill than the people who don't smoke. The people who don't inhale tobacco have 1.07 more risk of being ill than the people who don't smoke.

How Should I Present the Results of Logistic Regression in a Slide or an Article? (Example, Please.)

Table 1: Risk factors for endometrial cancer.

Variables	β_i	Se β_i	$OR=e^{\beta_i}$	Confidence Interval (CI _{0.95})	p-value
Gallbladder disease (dichotomous variable)	1.150	0.446	3.158	[1.316 ; 7.578]	0.010
Estrogen (dichotomous variable)	1.718	0.517	5.576	[2.021 ; 15.383]	< 0.001
Duration of estrogen treatment (Continuous variable)	0.011	0.005	1.012	[1.002 ; 1.022]	0.017

Likelihood ratio chi square test: 46.58 with 3 degrees of freedom (p < 0.0001).

NOTE: For a less technical audience, the beta coefficient (β_i) and the standard error of the beta coefficient ($Se\beta_i$) can be omitted.

References

Breslow NE, Days, NE. Statistical methods in cancer research, Vol. 1: The analysis of case-control studies. Lyon, France: IARC Scientific Publication No. 32; 1981.

Hosmer DW, Lemeshow S. Applied logistic regression (Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics Section). New York: John Wiley; 1989.

Kleinbaum DG, Kupper LL, Morgenstern H. Epidemiologic research: Principles and quantitative methods. New York: Van Nostrand Reinhold; 1982.

Kleinbaum DG, Kupper LL, Muller KA. Applied regression analysis and other multivariate methods. Second edition. Boston: Duxbury Press; 1987.

Kleinbaum DG. Statistics in the health sciences: Logistic regression. New York: Springer-Verlag; 1994.

Mack TM, Pike MC, Henderson BE, Pfeffer RI, Gerkins VR, Arthur M, Brown SE. Estrogens and endometrial cancer in a retirement community. N Engl J Med 1976 Jun 3;294(23):1262-7.

Kaplan-Meier Survival Analysis



The KMWin Program

Authors: Cecile Delhumeau, Andrew Dean

What is the objective of the Kaplan-Meier method?

In clinical epidemiology, particularly in the study of usually fatal chronic diseases, the measurement of patient survival has become an important criterion in evaluating the effectiveness of therapeutic modalities.

The objective of the Kaplan-Meier (KM) methodology is to estimate the probability of survival of a defined group at a designated time interval (conditional probability). KM uses a non-parametric survival function for a group of patients (in other words their survival probability after the time t) and therefore does not make assumptions about the survival distribution.

What is KM good for?

Each time a survival study is done (e.g., a follow-up study, a clinical trial, or a study of the occurrence of an event over time), the KM methodology could be used to estimate the probability of survival over a given time period. “Survival” means that the event of interest has not occurred. The event can be death, a complication of a treatment, or other defined adverse event. KM therefore provides an estimate of being free of the event at time t . Conversely, “1 minus the probability of being free of the event at time t ” is the probability of having the event at time t .

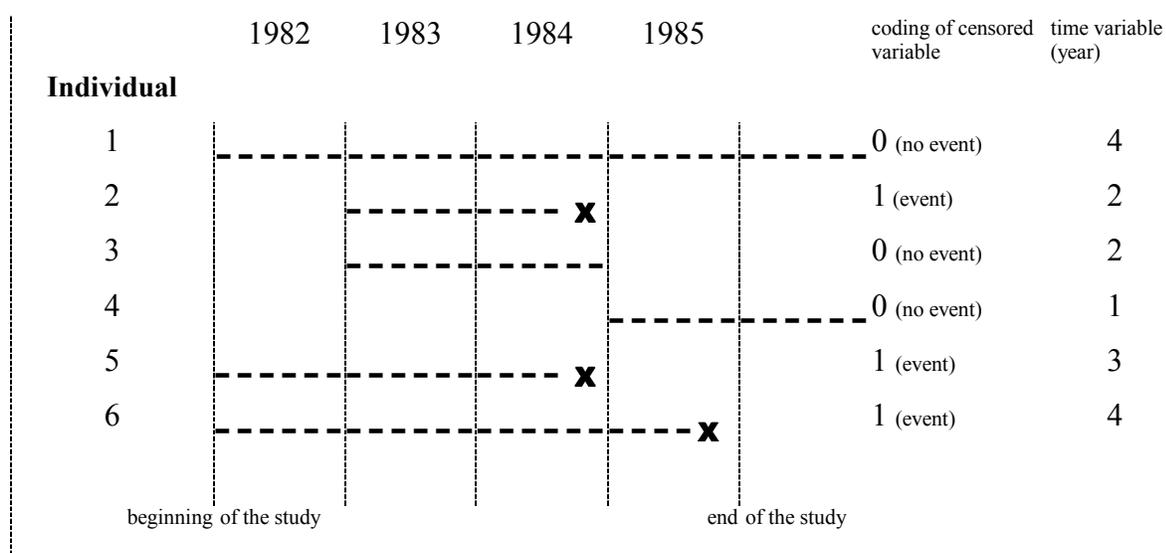
What is the variable to be studied?

The variable to be studied is the time delay until the occurrence of an event (death, disease, treatment outcome, etc.). This time delay corresponds to survival duration (the difference between the beginning study date and the event date).

What distinguishes survival analysis from most other statistical methods is the presence of “censoring” for incomplete observations. In a study of survival following two different treatment regimens, for example, analysis of the trial typically occurs well before all the patients have died. For those still alive at the time of analysis, the true survival time is known only to be greater than the time observed to date. Such an observation is said to be “censored.” There are two other sorts

of incomplete observation: the “lost to follow-up” (patient missing during the study duration) or the appearance of an event other than the event being studied. These observations are also considered censored. For survival analysis, the censored variable, the time variable, the units of time (day, month, year), and the group of patients (if studying the effect of a treatment) must be specified. The time variable is numeric. The censored variable is coded: “1” if the patient experiences the event (uncensored data), “0” if the event is not known to occur (censored data). Survival data is often presented using a “+” for the censored observation, so that a set of times might be 8, 11+, 14, 2, 36+, etc.

Table 1: Coding of censored variable for 6 patients with bladder cancer.

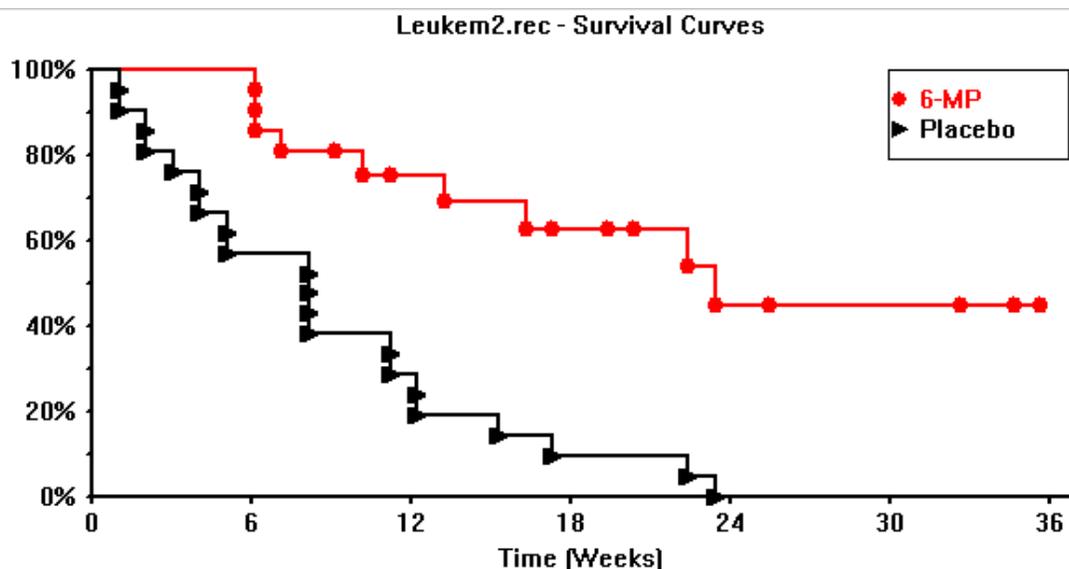


The censored variable has the value of “0” for individuals 1, 3, and 4, and “1” for individuals 2, 5, and 6.

What is the shape of the KM survival function?

The KM survival function is a decreasing series of straight line steps, constant between two consecutive death times, with a step for each time of observed death. This function is not defined after the last observation if this observation is censored.

Here is an example of the shape of the KM survival function:



How can I interpret the result of the KM curve?

If the survival rate for a group of patients 5 years after the beginning date of the study is 0.8, this means these patients have, on the average, 80 chances out of 100 of remaining alive after 5 years. In this example, the occurrence of an event is “death,” but we could also describe other events, for example, the occurrence of complications during medical treatment.

How does Epi Info 2000 compute the KM estimator?

In the first step Epi Info sorts the records by time, t_i , then for each time interval t_i up to t_{i+1} (but not including) it counts:

- * the number of deceased patients d_i at time t_i ,
- * the number of censored patients c_i at time t_i ,
- * the number of risk patient n_i (number of patients living just before t_i)

$$n_i = n_{i-1} - c_{i-1} - d_{i-1}$$

After all the steps, it is possible to compute the KM estimator:

$$S = \prod \left(1 - \frac{d_i}{n_i} \right) \quad (\text{“}\Pi\text{” means “product of”})$$

The example provided in this documentation uses the Leukemia study (Freireich, 1963), a study about the remission time delay for patients with leukemia who are given different treatments. This example is based on a randomized clinical trial to evaluate if patients assigned to treatment with 6-mercaptopurine (6 – MP) would fare better than untreated (Placebo) patients.

The project file is provided with Epi Info 2000 as leukem2.kmp

Key variables:

- * time delay (in weeks)
- * censored variable (“0” for censored individual, “1” for uncensored individual)
- * group of patients (6-MP or Placebo)

KM estimator for the group 6-MP

Time in Weeks	S(t)	Standard Error	Cumulative Failures	At Risk
6.00	0.95238	0.04647	1	20
6.00	0.90476	0.06406	2	19
6.00	0.85714	0.07636	3	18
6.00+			3	17
7.00	0.80672	0.08694	4	16
9.00+			4	15
10.00	0.75294	0.09635	5	14
10.00+			5	13
11.00+			5	12
13.00	0.69020	0.10681	6	11
16.00	0.62745	0.11405	7	10
17.00+			7	9
19.00+			7	8
20.00+			7	7
22.00	0.53782	0.12823	8	6
23.00	0.44818	0.13459	9	5
25.00+			9	4
32.00+			9	3
32.00+			9	2
34.00+			9	1

35.00+			9	0
--------	--	--	---	---

+ denotes censored patient (c_i)

Where:

Time is the day of observation

$S(t)$ is the survivor function $\Rightarrow \Pr(T > t)$

Standard Error is the standard error of $S(t)$

Cumulative Failure is the number of deceased patients (d_i)

At Risk is the number of at-risk patients (n_i)

The patients in the 6-MP group have, on the average, 44.8 chances out of 100 of remaining alive after 23 weeks.

Quartile	Estimate
75th	13.00
50th	23.00

75 % from the sample live on the average 13 weeks

50 % from the sample live on the average 23 weeks

Mean Survival Time (MST) in Days

Method	MST	Limited To	Standard Error
Last Obs.	23.287	35.00	2.9990
Cutpoint	16.117	20.00	1.3518
Comparison	17.909	23.00	1.6474

KM estimator for group Placebo :

Time Weeks	S(t)	Standard Error	Cumulative Failures	At Risk
1.00	0.95238	0.04647	1	20
1.00	0.90476	0.06406	2	19
2.00	0.85714	0.07636	3	18
2.00	0.80952	0.08569	4	17

3.00	0.76190	0.09294	5	16
4.00	0.71429	0.09858	6	15
4.00	0.66667	0.10287	7	14
5.00	0.61905	0.10597	8	13
5.00	0.57143	0.10799	9	12
8.00	0.52381	0.10899	10	11
8.00	0.47619	0.10899	11	10
8.00	0.42857	0.10799	12	9
8.00	0.38095	0.10597	13	8
11.00	0.33333	0.10287	14	7
11.00	0.28571	0.09858	15	6
12.00	0.23810	0.09294	16	5
12.00	0.19048	0.08569	17	4
15.00	0.14286	0.07636	18	3
17.00	0.09524	0.06406	19	2
22.00	0.04762	0.04647	20	1
23.00	0	0	21	0

+ denotes censored patient (c_i)

Where :

Time is the week of observation

$S(t)$ is the survivor function $\Rightarrow \Pr(T > t)$

Standard Error is the standard error of $S(t)$

Cumulative Failure is the number of deceased patients (d_i)

At Risk is the number of risk patient (n_i)

The patients in the Placebo group have, on the average, 0 chances out of 100 of remaining alive after 23 weeks.

Quartile	Estimate
75th	4.00
50th	8.00
25th	12.00

75 % from the sample live on the average 4 weeks

50 % from the sample live on the average 8 weeks

25 % from the sample live on the average 12 weeks

Mean Survival Time (MST) in Days

Method	MST	Limited To	Standard Error
Last Obs.	8.667	23.00	1.4114
Cutpoint	8.429	20.00	1.3896
Comparison	8.667	23.00	1.4114

Summary Table for the Data:

Group	Total	Fail	Cens.	%Fail	%Cens.	FDR(*)	Lower	Upper
6 – MP	21	9	12	42.86	57.14	0.0251	0.0153	3.7213
Placebo	21	21	0	100.00	0.00	0.1154	0.0250	3.9093
TOTAL	42	30	12	71.43	28.57	0.0555	0.0183	3.7851

(*) Failure Density Rate (#Failures / Person * Time).

Where :

Total is the sample size

Fail is the number of deceased patients

Cens is the number of censored patients

%Fail is the percentage of deceased patients

%Cens is the percentage of censored patients

FDR is the Failure Density Rate (#Failures / Person * Time).

Lower and Upper are the limits of the 95 % Confidence Interval of FDR

How should I present the results of KM in a slide or an article?

The presentation of results for a KM analysis might be a graph like the one shown previously for each group of patients with 95% confidence interval at each time *t* (precision of the survival estimation) and a table with the survival estimates at 1, 2, and 5 years (all depend on the disease being studied).

How should I evaluate whether or not KM curves for two or more groups are statistically equivalent?

When we state that two KM curves are “statistically equivalent,” we mean that, based on a testing procedure that compares the two curves in “some overall sense,” we do not have evidence to indicate that the true (population) survival curves are different.

The software uses the log-rank test, a large-sample chi square test that uses as its test criterion an overall comparison of the KM curves being compared. This (log-rank) statistic, like many other statistics used in other kinds of chi square tests, makes use of observed versus expected cell counts over categories of outcomes.

The software also uses the generalized Wilcoxon test, a non-parametric test. This might be used, for example, to test for a significant difference in treated and untreated patients. This test assumes that the death rate is constant over time (the software does a logarithm transformation of the KM survival curve).

A p-value for the log-rank test <0.05 suggests a difference in survival between the two groups.

Example :

Summary Table for the Data:

Group	Total	Fail	Cens.	%Fail	%Cens.	FDR(*)	Lower	Upper
6 – MP	21	9	12	42.86	57.14	0.0251	0.0153	3.7213
Placebo	21	21	0	100.00	0.00	0.1154	0.0250	3.9093

TOTAL 42 30 12 71.43 28.57 0.0555 0.0183 3.7851

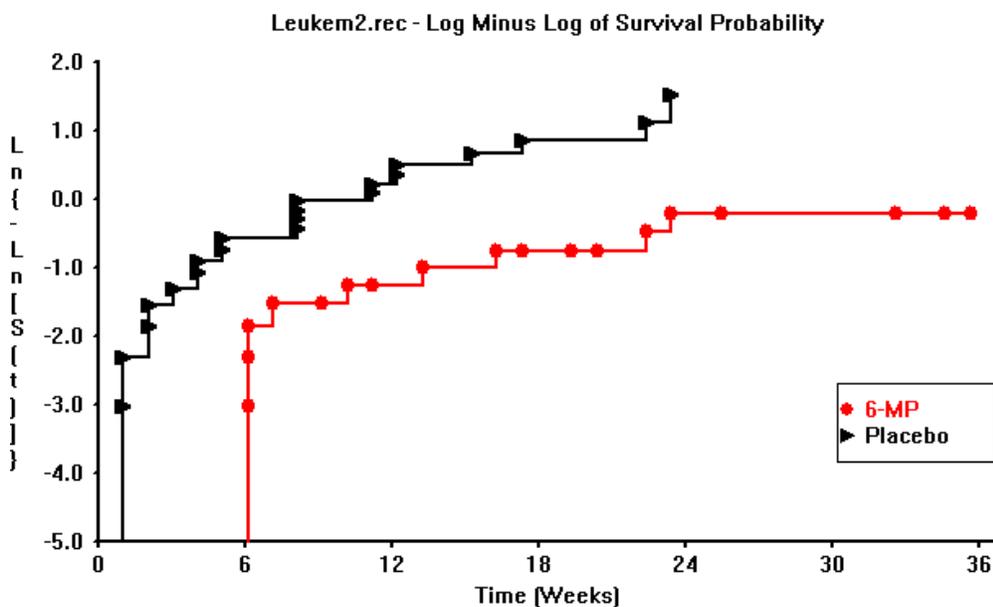
(*) Failure Density Rate (#Failures / Person * Time).

Test Statistics

Test	Statistic	DF	p-Value
Cox-Mantel (Log-Rank)	16.7929	1	0.00004
Generalized Wilcoxon (Breslow)	13.4579	1	0.00024

The two p-values are <0.05. We conclude that patients with the treatment 6-MP live longer than patients with the placebo treatment.

Testing the assumption that death rate is constant over time in this example:



References

Freireich EJ, Gehan E, Frei E. The effect of 6-mercaptopurine on the duration of steroid-included remissions in acute leukemia: a model for evaluation of the other potentially useful therapy. *Blood* 1963; 21: 699-716.

Kleinbaum DG. *Statistics in the health sciences: Survival analysis*. New York: Springer-Verlag; 1996.

Appendix: Resources for Creating Public Health Maps

Web References

The following links are provided as a guide to exploring the dynamic field of health and geographic information science. Outside links are not supported by CDC, but are listed for their related content.

- [Overviews](#)
- [Geographic Boundaries](#)
- [Data](#)
- [Geostatistics / Spatial Analysis](#)
- [Global Public Health GIS Projects](#)
- [Conferences and Symposia](#)
- [Software](#)
- [Consultants](#)

Overviews

- [GIS In Depth](#), Charles Sturt University, Division of Information Technology, Australia.
- [Information Architecture White Paper IA-5A10: Geographical Information Systems \(GIS\)](#), Los Alamos National Laboratory, July 1997.
- [Improving Public Health Through Geographical Information Systems An Instructional Guide to Major Concepts and Their Implementation](#), Web Version 1.0, December 1997. By Gerard Rushton at the University of Iowa.
- [Public health geographic information systems \(GIS\) news and information: 1994-1997](#), by Charles Croner (January 1998). CDC National Center for Health Statistics Cognitive Methods Working Paper Series Number 23. Can be ordered online.
- [Visualizing Health Statistics](#), CDC National Center for Health Statistics and the Pennsylvania State University.

Geographic Boundaries

Use these resources to locate *.shp boundary files for use with any ESRI-based GIS software, including Epi Map 2000.

Geospatial Link Repositories

- [ESRI ArcData Online: GIS Data on the Web](#)
- [ESRI Packaged Geographic Data Sets -- Global](#)

- [ESRI U.S. State Plane Zones Data Set](#)
- [Federal Geographic Data Committee](#)
- [Geospatial Data Clearinghouse Entry Points](#)
- [GIS Jump Station – Domestic Federal Sites](#)
- [GIS Jump Station – International Gov't Sites](#)
- [GIS Resources for the Federal Government](#)
- [National Atlas of the United States](#)
- [National Geodetic Survey Products and Services](#)
- [National Imagery and Mapping Agency \(NIMA\)](#)
- [US Census Bureau TIGER Mapping Service](#)
- [US Geological Survey National Mapping Information](#)
- [US Mortality Data Shapefile Download](#)

- [Alexandria Digital Library, University of California, Santa Barbara](#)
- [Center for Advanced Spatial Technologies](#)
- [Center for Geographic Information Systems Research, Georgia Institute of Technology College of Architecture](#)
- [Computers in Teaching Initiative \(CTI\) Medical Geography, University of Leicester, UK](#)
- [GDT's premier street network database, Dynamap/2000®](#)
- [Geographic Information Systems](#)
- [GIS-Expertisecentrum Rijksuniversiteit Groningen](#)
- [GIS Resource Centre](#)
- [Harvard Map Collection](#)
- [Health Geographic Resources](#)
- [Institut für Geographie Department of Geography Salzburg, Austria](#)
- [Just Another Medical Geography Page](#)
- [MAGIC: Map and Geographic Information Center](#)
- [NASA Langley Geographic Information Systems](#)
- [National Cartography and Geospatial Center \(NCGC\)](#)
- [National Center for Geographic Information and Analysis, University of California at Santa Barbara](#)
- [Northwestern University Library Government Publications and Maps](#)
- [PennState University Libraries Maps and Geography](#)
- [Sandia National Laboratories GIS WWW Sites, by Topic](#)
- [Spatial Odyssey GIS Literature Database](#)
- [Terra Data Links to Mapping Related Sites](#)
- [Universiteit Utrecht Faculty of Geographical Sciences GLOBIS: Nice Geography Sites](#)
- [Universiteit Utrecht Oddens's Bookmarks: The Fascinating World of Maps and Mapping](#)
- [University of Edinburgh Department of Geography](#)
- [University of Virginia geostat: Geospatial and Statistical Data Center](#)
- [US Fish & Wildlife Service Region 5 GIS Lab Home Page](#)

- [US Geological Survey GTOPO30 Global 30 Arc Second Data Set](#)
- [US Geological Survey Map-It: Form-based Simple Map Generator](#)

Geospatial – USA State GIS Data Centers

- [Arkansas Interactive Mapper, Center for Advanced Spatial Technologies](#)
- [Colorado ELBERT Geographic Information Coordinating Committee](#)
- [Connecticut Dept of Environmental Protection DEP Store: The Maps and Publications Office of the Natural Resources Center](#)
- [Georgia Spatial Data Infrastructure](#)
- [Massachusetts Geographic Information System: MassGIS](#)
- [Maine Office of GIS](#)
- [New Hampshire GRANITNet](#)
- [New Jersey DEP's GIS web page](#)
- [New Mexico GIS Directory and Jump Station](#)
- [Oregon State Service Center for Geographic Information Systems](#)
- [Rhode Island Geographic Information System](#)
- [Texas Natural Resources Information System](#)
- [Vermont Geographic Information System](#)
- [Virginia TIGER/Line Data Browser](#)

Global Positioning System (GPS)

- [Global Positioning System Resources – University of Iowa](#)
- [Trimble - All About GPS](#)
- [Trimble Community Base Stations Index](#)
- [US Coast Guard Maritime Differential GPS Service](#)
- [USDA Forest Service GPS Page](#)
- [Washington State GPS Reference Base Station Information](#)

Images

- [Aerial Photos, USDA Farm Service Agency](#)
- [Microsoft TerraServer World Imagery Database](#)
- [SPIN-2 Image Library: Worldwide Coverage](#)

Data Sources

These sites provide numerators and denominators for data to be mapped in your public health projects.

Demographic

- [Columbia University Center for International Earth Science Information Network \(CIESIN\) Socioeconomic and Applications Data Center \(SEDAC\)](#)
- [EUROSTAT: Statistical Office of the European Communities](#)
- [Population Council](#)
- [Population Reference Bureau](#)
- [United Nations Population Division](#)
- [United Nations Statistics Division](#)
- [U.S. Census Bureau](#)
- [U.S. Census State Data Centers](#)
- [U.S. Census Bureau International Programs Center](#)
- [World Bank Group Development Data](#)

Disease Surveillance

- [Atlas of US Mortality](#)
- [EuroSurveillance: European Communicable Disease Bulletin](#)
- [Pan American Health Organization](#)
- [Public Health Laboratory Service – United Kingdom](#)
- [US Centers for Disease Control and Prevention](#)
- [US Department of Health and Human Services](#)
- [World Health Organization](#)

Environmental Data

- [BioNet Environmental Information System](#)
- [Catalina Island Conservancy Intranet Ecological Management Programs GIS Submenu](#)
- [Center for Marine Conservation](#)
- [Compass Informatics, Irish Environmental Protection Agency's National Freshwater Quality Database](#)
- [Environment Australia Online](#)

- [Environmental Measurements Laboratory Databases](#)
- [Florida Marine Research Institute](#)
- [Florida Statewide Ocean Resource Inventory](#)
- [Geospatial Technology Activities at the Patuxent Wildlife Research Center](#)
- [Groundwater.Com](#)
- [Holt's GAP analysis and GIS links](#)
- [ICE MAPS Interactive California Environmental Management, Assessment, and Planning System](#)
- [InfoRAIN: Bioregional Information System for the North American Rainforest Coast](#)
- [MALSAT: Environmental Information Systems for Malaria](#)
- [National Center for Ecological Analysis and Synthesis](#)
- [National Center for Environmental Assessment](#)
- [National Center for Environmental Health Sciences](#)
- [Pacific Northwest National Laboratory Geographic Information Systems](#)
- [RAY MINE PILOT STUDY: Advanced Monitoring of Hazardous Waste Sites: Discrimination and Screening of Problem Mine and Extractive Industry Wastes](#)
- [University of Miami Geocore GIS Research Facility](#)
- [US Environmental Protection Agency](#)
- [US Fish & Wildlife Service Environmental Contaminants Program](#)
- [US National Oceanic and Atmospheric Administration Coastal Services Center](#)

Geostatistics And Spatial Analysis

- [AI - GEOSTATS](#)
- [Center for Spatial Analysis Technologies](#) (Georgia Tech and USGS)
- [Geographical Information Science Tutorials in the World Wide Web, University of Hong Kong](#)
- [Geovariances: The Universe of Geostatistics](#)

Global Public Health Gis Projects

Use these resources to locate examples of use of GIS in public health projects. You may be inspired to create similar maps with your own data.

Online Atlases

- [American FactFinder](#), U.S. Bureau of the Census
- [An Atlas of Injury Death in Australia 1990-1992](#)
- [Anatomy of an Epidemic](#). An elementary example of student work.
- [ArcData Online: GIS Data on the Web](#)
- [The Atlas of Mortality in Europe](#). A WHO/UN/Statistics Netherlands/CIVM collaboration with interactive maps by year, gender, and ICD9 code.

- [Atlas of United States Mortality](#). A CDC NCHS Website with downloadable Acrobat (*.pdf) files with various causes of death. Also, some maps available for online viewing through your browser.
- [Demographic Data Viewer, CIESIN/SEDAC](#)
- [Mortality Atlas of Cancer and Other Causes, Spain 1975-1986](#)
- [NASA Ocean Color Data and Resources: Health Applications of CZCS Data](#)
- [The Sentinel System, INSERM, French Ministry of Health.](#)
- [Social Health Atlas of South Australia](#)
- [US Environmental Protection Agency Envirofacts Warehouse Maps on Demand](#)
- [WHO Communicable Disease Surveillance and Response](#)
- [WHO Health for All Database Queries for Europe and NIS](#)
- [Women and Heart Disease: An Atlas of Racial and Ethnic Disparities in Mortality](#), CDC Cardiovascular Health Branch and Office for Social Environment and Health Research at West Virginia University

Conferences And Symposia

- [International Health Geographics Conference 2000](#), Washington, D.C., March 17 - 19, 2000.
- [First International Health Geographics Conference](#), Baltimore, October 15-18, 1998.
- [1998 GIS in Public Health Conference](#), San Diego, August 18-20, 1998.
- [GIS and the Health Sector '98](#), University of Melbourne, Australia, June 10, 1998
- [GIS and the Health Sector '97](#), University of Melbourne, Australia, July 9, 1997
- [GIS for Health and the Environment](#), Proceedings of an International Workshop, Colombo, Sri Lanka, September 5-10, 1994.

Software

GIS software packages that you can either purchase or download for free.

- [CaveTools for ArcView GIS](#)
- [Claritas](#)
- [Community 2020](#) HUD Community Planning Software
- [DismapWin](#) freeware developed by Dr. Peter Schlattmann at the Institut für Soziale Medizin und Medizinische Psychologie
- [EpiMap](#), freeware developed by CDC and WHO.
- [ESRI](#)
- [GAEA: Geoscience and Engineering Applications](#)
- [Healthmapper](#) is the software of HealthMap, a joint WHO/UNICEF Programme.
- [IDRISI](#), produced and supported by the Department of Geography at Clark University.
- [Lakes Environmental Software: Air Modeling & Risk Assessment](#)
- [MapInfo](#)

- [Map Maker](#) (Scotland, UK)
- [PopMap](#), freeware developed by the University of Hanoi for the United Nations Statistics Division.
- [SaTScan](#), public domain software for calculating spatial, temporal and space-time scan statistics, developed at the National Cancer Institute.

Gis Consultants With Experience In Public Health

CDC does not endorse or recommend specific vendors. Commercial firms experienced with public health GIS projects for clients such as U.S. state health departments are invited to submit links for placement in this section.

- [Applied Geographics, Inc.](#)
- [Scientific Technologies Corporation](#)
- [Silent Spring Institute](#)

Index

- 2 x 2 x S Tables..... 221
- A Nutritional Anthropometry Program..... 75
- ABS..... 157, 163
- active data set..... 131
- ActiveX Modules..... 189
- AGE , Weight = AGESUMMARY.COUNT..... 209
- Analysis 13, 15, 17, 18, 20, 21, 24, 25, 26, 27, 29, 32, 36, 41, 43, 44, 48, 49, 52, 53, 55, 56, 57, 58, 63, 67, 68, 69, 70, 73, 75, 84, 89, 92, 93, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 157, 159, 177, 182, 191, 192, 196, 197, 203, 204, 205, 206, 233, 235, 238, 242, 245, 257, 264, 299
- Cleaning, Transforming, and Analyzing Data..... 24
- Analysis (.PGM) Programs..... 20
- ANALYSIS program..... 67, 84, 90, 92, 93
- Analysis.Exe..... 92
- Analyzing
- A Single Numeric Variable—The MEANS
- Command..... 236
- All Types of Data—The LIST Command..... 207
- Numeric Variable Across Groups..... 237
- Summary Data With a COUNT..... 208
- Text, Numbers, and Dates—The FREQ Command
- 207
- Text, Numbers, or Dates..... 212
- Analyzing Common Types of Data..... 207
- AND 98, 120, 128, 138, 143, 146, 157, 159, 160, 161
- ANOVA..... 125
- ANOVA test (Analysis)..... 68
- Anthropometric Calculation Program..... 83
- Anthropometric Errors..... 83
- anthropometric indices..... 80, 81, 82, 83, 87
- Anthropometric Software..... 86
- Anthropometry..... 80
- Arithmetic (+ - * / ^ MOD)..... 158
- arrow keys..... 99
- Assign..... 43, 96, 97
- Assistance..... 1, 3
- asterisk..... 94
- Automatic coding..... 66
- AUTOSEARCH..... 90, 99, 100
- Backing Up Data Files..... 58
- Beaton..... 86, 87
- Become acquainted with the Epi Info 2000 programs
- 13
- Beep..... 100
- Button(s)..... 101
- Buzina..... 87
- Cancel Sort/Select..... 102
- CAT by CHD..... 214, 222
- centimeters..... 82, 83, 85
- CHECK..... 138
- Check (.CHK) Files..... 19
- CHECK Programs..... 91
- Checking and Controlling Data Entry..... 34, 65
- citation**..... 1
- CITY..... 122
- CLEAR..... 30, 90, 102
- Closeout..... 103
- Cmd..... 103
- CMD..... 103
- Command Blocks..... 104
- Commands in the Epi Info Programming Language..... 89
- Comments (*)..... 105
- Common errors..... 84
- Components..... 13
- computer virus..... 2
- Conditional logistic regression..... 236
- confidence limits..... 2, 112
- Confounding and Interaction..... 221
- Converting
- Approaches..... 19
- Epi Info 6.xx Systems to Epi Info 2000..... 19
- Systems with Related Files..... 20
- When is it wise?..... 19
- cross tabulation..... 124, 148
- Custom Installation..... 13
- Cut, Copy, and Paste fields (MakeView)..... 64
- cutoff value..... 81, 82
- Data (.REC) Files..... 19
- Data Entry Features..... 17
- Data Management..... 20, 49, 67
- Date and Time Functions..... 167
- DAY..... 157, 168, 169

DAYS.....	157, 168, 169	Epi Info 2000 Views	49, 57, 181
dBASE	138	Epi Info 6 for DOS.....	15
dBASE File	44	Epi Info Worldwide Discussion Group.....	3, 8
Dbase, Paradox, FoxPro, Excel, and Microsoft		Epi Map... 1, 2, 7, 8, 9, 10, 13, 18, 22, 27, 30, 70, 192	
Access files	67	EPI MAP	70
Define.....	40, 105	Epi2000.Exe.....	91
DEFINE	105, 131, 150	Epimap.Exe.....	94
defined variables	152	Equations for 2x2 Tables	216, 220
Defining a New Variable	42	Equations for Statistical Tests for Overall	
Delete an existing data table	64	Association.....	232
Desktop Applications.....	11	Evaluations, and Specific Conditions	73
Dialog.....	100, 107, 108, 109, 178	Evans County data listing for CAT and CHD.....	214
Dibley.....	86, 87	Exact statistics (Analysis)	68
Differences		Examples Included With the System	73
Before and After	245	EXECUTE(LINKTO or ACTIVATE).....	110
Division.....	86	EXISTS	157, 177, 178
Downloading from the Internet.....	9, 11	Exit (Quit)	111
Draper	137	EXP	157, 162
Drive and Path.....	13	expression	97, 143
drivers	2	File or table	
EndBefore	109	Creating with the WRITE Command.....	44
Enter13, 21, 24, 29, 30, 32, 33, 35, 45, 46, 50, 52, 55,		FILEDATE	157, 178
57, 61, 64, 65, 66, 67, 68, 69, 70, 91, 92, 93, 94,		Finding Records.....	36
98, 99, 101, 105, 106, 108, 112, 121, 127, 129,		FINDTEXT	157, 174, 175
130, 135, 138, 152, 182, 194, 299		FLAG	83, 84
Data Entry Program for Windows.....	24	FLAG field.....	84
ENTER.....	99	flags.....	84
ENTER and CHECK	138	Fleiss	253, 254
ENTER program 21, 29, 31, 35, 52, 57, 66, 90, 92,		FORMAT.....	157, 166, 167, 173, 174, 175
110		format of the tables	149
Enter.Exe.....	92	FREQ	112, 113
ENVIRON	157, 177	Frequencies	41, 138, 208, 209
Epi 2000 Menu		FREQuencies	41, 68, 112, 125
Adding or changing a button.....	62	Frequently-Asked Questions.....	61
Adding or changing a menu item.....	61	Functions and Operators	34, 89, 157
Changing the image	61	Geographic Information System (GIS)	7, 18, 27
Developing a new menu (.MNU) file.....	61	Geographic variable	70
Help system.....	63	Getting Acquainted with Epi Info 2000	13
Programming the response.....	62	Getting Acquainted With the Programs	31
Setting the working directory.....	62	GLOBAL	131
Turning buttons on or off.....	62	Goldsby	87
EPI 2000 Menu	61	Gorstein.....	86, 87
Epi 6 and Epi 2000 Systems		GOTO	16, 35, 65, 90, 102, 113, 114
Helping them to Coexist	20	Graph	114
Epi Info 2000 Commands	90	Graph (Analysis).....	69
Epi Info 2000 Data Tables	50	Graphing	18
Epi Info 2000 Menu	13, 19, 29, 197	Group	86

Groups of Variables 36
 Growth Reference Curves 80, 82, 85
 growth velocity 80
 Guided Tour 22, 31, 32, 35, 40, 45, 46, 48, 63, 65, 67,
 68, 70
 Analysis Program 40
 Enter Program 35
 Epi Info 2000 Menu 31
 Epi Map Program 46
 MakeView Program 32
 VisData Program 48
 GUIDED TOUR 13
 HEADER 114, 123
 HELP 42, 63, 90, 116, 117, 197, 198, 199
 HIDE 117
 HIDE, UNHIDE 117
Hotline 1
 HOUR 157, 167, 170, 171, 172, 173
 HOURS 157, 170, 171, 172, 173
 HOUSE 138
 How to
 Assess arm circumference (NutStat) 69
 Assess height and weight (NutStat) 69
 Cancel a previous selection (Analysis) 67
 Change colors or patterns (Epi Map) 70
 Change the Prompt, Name, or Type of a Field
 (MakeView) 64
 Change the values displayed in Yes/No fields
 (Enter) 66
 Choose a boundary layer (Epi Map) 70
 Combine data in more than one table (Analysis) 68
 Compact a database (VisData) 71
 Convert a file from Epi Info, Version 6
 (MakeView) 65
 Create a Related Form/View (MakeView) 64
 Delete or rename a table, field, or file (VisData) 71
 Display a map (Epi Map) 70
 Display Rates (Epi Map) 70
 Display values from a data file in a map using
 Analysis 70
 Display values from a data file using Epi Map (Epi
 Map) 70
 Do Skip Patterns (MakeView) 65
 Group fields together as a functional unit
 (MakeView) 64
 Insert a background picture or color (MakeView)
 64

Insert a grid (table) in a view (MakeView) 64
 Line up fields (MakeView) 64
 Make a view 63
 Make a view of an existing database (MakeView)
 65
 Move among parent and related forms (Enter) ... 67
 Move from record to record (Enter) 66
 Move Related Field buttons (MakeView) 65
 Move the Legend (Epi Map) 70
 Obtain an age or duration (MakeView) 65
 Open or create a data file (NutStat) 69
 Prepare a data file for use with a map (Epi Map) 70
 Process data from a questionnaire study (Analysis)
 68
 Process only certain columns (variables)
 (Analysis) 68
 Read a file format and write to another (Analysis)
 67
 Repair a damaged database (VisData) 71
 Save a page (Enter) 67
 Save a record (Enter) 67
 See data in spreadsheet format (VisData) 71
 Select records (Analysis) 67
 Set up units of measurement (NutStat) 69
 Specify Legal Values (MakeView) 65
 Use Nutstat in combination with another data entry
 view (NutStat) 69
 How To 61
 IDNUM 113
 IEPI Modules 192
 IEPI Type Library 189
 IF Statement 42
 If Then Else 118
 IGNORE 126
 Importing, Exporting, Relating, and Accessing Data
 49
 Installation 11, 12, 22
 From CD-ROM 11
 Full 13
 Problems 14
 Installation by TSETUP.EXE 198
 Interaction 261, 263, 264, 269
 Internet 1
 Internet Features 17
 Kaplan-Meier
 Curve? 275
 Estimator? 275

- Method 273
- Presenting results 280
- Survival function? 274
- Two or more groups 280
- Kaplan-Meier Survival Analysis 120, 273
- Keller 87
- Kelly 87
- Kevany 87
- Key Features of Epi Info 2000 7
- kilograms 83, 85
- Kish 253, 255
- Kmsurvival 120
- KMWin Program 273
- L.A. Study
 - Endometrial cancer 265
- Language 198
- LANGUAGE.MDB in the Main Epi Info 2000
 - Directory 198
- Languages 201
- LEGAL 130
- LET 97, 131
- Limitations 82, 85
- Limitations and Problem Resolution 193
- Limitations of Growth Reference Curves 82, 85
- List 8, 102, 107, 108, 121
- LIST 121, 122, 146
- Lists 17, 41, 181, 184, 188
- LN 157, 162, 163
- LOG 14, 22, 157, 162, 163
- Logistic 8, 17, 25, 122, 203, 233, 234, 236, 257, 258, 261, 263, 269, 270, 271
- Logistic Regression 233, 257
 - Categorical Variables 260
 - Confounding assessment 264
 - Continuous Variables? 260
 - Improper Use 264
 - Model 261, 266
 - Outcome Variable? 258
 - Presenting Results 270
 - Terms 263, 269
 - Types of Variables 257
- Logistic Regression and Kaplan-Meier Survival
 - Analysis 17
- low anthropometric values 82
- MakeView.. 16, 19, 20, 21, 23, 24, 27, 29, 32, 35, 51, 56, 57, 58, 63, 64, 65, 66, 70, 79, 89, 90, 92, 98, 101, 104, 121, 127, 130, 139, 157, 182, 188, 194, 197
- Questionnaire and Form Designer 23
- Makeview.Exe 91
- Making a Database 35
- Manual** 1
- Map 2, 7, 18, 22, 27, 70, 94, 123, 192
- Maps and Graphs 24
- Marks 2
- Match 124, 204, 234, 235
- Matched Cases and Controls—The MATCH Variable 234
- McAfee 2
- Means 125, 206, 237, 247, 248
- MEANS** 125, 126
- MEANS command 125
- Means of AGE 237
- Means of DIFFERENCE 248
- Menu 11, 22, 23, 62, 63, 89, 91, 93, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 135, 136, 137, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 195, 197, 299
- Menuitem 127
- Merge 128
- MetaData Table 183
- Microsoft Access Databases 49
- Microsoft Access File Format 16
- Microsoft Access Links 50
- Microsoft Word 2
- MINUTE 157, 167, 170, 171, 172, 173
- MINUTES 157, 170, 171, 172, 173
- Model 261, 265
- Modifying Values Within Expressions 157
- MONTH 157, 168, 169
- MONTHS 157, 168, 169
- More than one file 138
- Moving From Record to Record 36
- Multiple Linear Regression 251
- Mustenter 129
- MUSTENTER 129, 130
- Next Steps 48
- Nieburg 86, 87
- NOENTER 117, 135
- normally distributed 82

NOT98, 112, 113, 120, 121, 122, 152, 157, 159, 160, 161, 177, 198

Notes on the Output Library in Analysis 152

Numeric Functions..... 162

NUMTODATE 157, 166, 167, 177

NUMTOTIME 157, 166, 167

Nutritional Anthropometry 45

nutritional deficiency 80

Nutstat

 As Part of an Epi Info 2000 Questionnaire View 79

 Configuring 76

NutStat 21, 26, 27, 30, 45, 46, 69, 75, 79, 80, 93

 Anthropometric Calculator..... 26

 The Nutritional Anthropometry Program..... 30

Nutstat.Exe..... 93

ODBC database such as SQL Server or Oracle (Analysis)..... 67

ODBC File Access

 Setting Up a DSN Link 55

Odds Ratio, Relative Risk, Chi Square and Fisher

 Exact test..... 68

Opening Another View 36

Operators..... 157, 158, 159

OPERATORS 98

OR..... 32, 62, 98, 118, 157, 159, 160, 161, 213, 216, 219, 224, 227, 230, 231, 233, 235, 236, 258, 261, 262, 264, 266, 268, 269, 270

Oswego study

 acute gastrointestinal outbreak..... 265

Other Anthropometric Software and Sources of Information 86

Outbreak Investigation..... 73

Overall Association..... 233

Overview..... 7, 61, 80, 89, 157, 181, 203

Parameter Estimates and Statistical Tests for

 Interaction 230

Parameter Estimates for a 2x2 Table 219

Pascal 2

percent of median..... 80, 81, 84

percentiles 80, 81, 84

Permanent variables 63

Picture 129, 196

Population data (Analysis)..... 69

Popup 130

PROCESS 112

Program Functions 21

Programmers 1

Programs 21

Proportions

 Confidence Limits for 210

 public domain..... 1, 7, 199

Questionnaire Features..... 16

R x 2 and R x 2 x S Tables..... 233

R x C Table 233

Range 130, 131

Rapid Tour 29

Rapid Tour For Experts In a Hurry 29

Read 19, 52, 57, 59, 67, 102, 114, 128, 131, 135, 138, 194

READ..... 112, 131, 138

READ ONLY..... 17, 19, 90, 135

READING a View in Analysis 41

Recode 136

RECODE 131

RECODE Command 44

References..... 9, 87, 183, 255, 271, 282

Refugee

 A Group of Related Data Views 73

REFUGEE.MDB 44, 52, 54, 73

Regress..... 137

REGRESS..... 137

REGRESS Birthweight = Estriol 249

REGRESS Command 249

REGRESS SystolicBlood = AgeInDays Birthweight 252

Regression... 137, 203, 233, 236, 249, 250, 252, 257, 258, 261, 263, 264, 269, 270

Regression Model 261

Relate 52, 57, 137, 139, 185

RELATE 137, 138

Related forms

 Conditional..... 66

Related Tables..... 50

Related Views, READING 44

Repeat 135, 140

RND..... 157, 164, 165

ROUND 157, 165

Routeout..... 140

RUN 141

Running Epi Info 2000..... 13

Runpgm..... 141

Sample 253

Sample size 253

Sample Size Calculations in Statcalc for DOS..... 253

Samples 73
 Screentext..... 96, 142
 SECOND 157, 167, 170, 171, 172, 173
 SECONDS 157, 170, 171, 172, 173
 Select.... 12, 13, 43, 55, 61, 62, 64, 65, 66, 67, 69, 70,
 71, 102, 143, 159
 SELECT..... 98, 112, 131, 143, 146
 SELECT Command 43
 Set 30, 62, 66, 69, 115, 144, 189
 SET 112, 126, 149
 SET IGNORE 126
 SET PERCENTS 149
 SET PROCESS 112
 Setbuttons..... 145
 Setpicture 146
 Setup, The Installation Program..... 22
 Shortcut to Epi Info 2000..... 13
 Simple Linear Regression 249
 SIN, COS, TAN 157, 162
 Single Table Analysis 215, 222, 223
 Smith..... 137, 147
 Sort..... 146
 SORT 121, 131, 146, 147
 Soundex 99, 147
 SOUNDEX 147
 Special Features of Check Commands..... 94
 Staehling 86, 87
 Standard Interface for Statistical Modules..... 18
 Standard table setup 225
 STATCALC 253
 Statistical Examples 73
 Statistical programmer's interface..... 189
 Statistical Tests
 Interpretation..... 221, 233
 Statistical Tests for Interaction for Stratified 2x2
 Tables..... 225
Statistics 1, 80, 203
 stratification 124
 Stratified 2x2 Tables 230, 232, 233
 Student's t test 125
 Student's t-Test (Analysis)..... 68
 subroutine..... 103
 SUBSTRING 157, 173, 174, 175
 SUBSTRINGS 97
 Sullivan 87
 SUMTABLES..... 148
 Surveillance

A Menu and Series of Data Input Forms..... 73
 Sysinfo 147
 System Functions 177
 System Requirements..... 8, 11
 System Requirements and Installation 11
 SYSTEMDATE 157, 177
 SYSTEMTIME 157, 167, 172, 177
 Tables... 24, 41, 51, 52, 102, 103, 132, 148, 182, 216,
 235
 Teaching Exercises 18
 Technical Support 8
 Text Functions 173
 The Broad Street Library of Statistics..... 181, 189
 Time Functions 170
 trademarks..... 2
 Translating the Epi Info 2000 Programs 197
 Translating the Manual and Help Files 198
 Translation 18, 197
 Fonts, Unicode, Right-to-Left 201
 Translation Features..... 18
 Translators 199
 Trowbridge..... 87
 TRUNC 157, 164, 165, 166, 175
 Turbo Pascal..... 2
 TXTTODATE..... 157, 173, 174, 177
 TXTTONUM 157, 173, 174, 177
 Type 33, 42, 56, 66, 69, 107, 108, 115, 149, 151, 182,
 183, 184, 196
 TYPE 149, 150
 U 125
 Undefine..... 150, 151
 UNHIDE 117, 120
 Uninstalling..... 11
 Upgrading to Epi Info 2000 from Epi Info for DOS. 8
 UPPERCASE..... 157, 174, 184
USA 1
 Using Epi Info 2000..... 21
 value of the variable 112
 variable name 112, 121, 138, 146, 149, 151
 variables 105, 112, 131, 138, 151, 152
 View Containing All Field Types Available in Epi
 Info 2000..... 184
 View Table..... 183
 Viewing Previous Results 42
 Views
 Structure of..... 181
 virus 2

Epi Info 2000

Index

VISDATA	71
VisData Program.....	27
VisData Utility for Viewing Data	30
Visual Basic . 2, 7, 12, 15, 18, 27, 176, 181, 189, 194, 201	
Waterlow.....	86, 87
What is Epi Info 2000?	7
WHO	1, 26, 75, 76, 79, 80, 81, 86, 87
WHO working group	87
Word Processor	26

World Health Organization	1, 80, 86
Write	151
WRITE RECFILE.....	152
XOR (eXclusive OR).....	160
YEAR.....	98, 117, 118, 157, 168, 169
YEARS	34, 97, 157, 167, 168, 169
Yes/No Fields.....	42
Yes/No or True/False Variables and 2x2 Tables...	213
Yip	87

Design a Questionnaire and Enter Data

Question or Prompt
Address

Font for Prompt

Field or Variable
Type: Text
Pattern: [None], Text(Uppercase), Multiline, Number, PhoneNumber, Date, Time

Field Name: Address

Required, Read Only, Soundex, Encrypt, Repeat Last Value, Index, Unique, Non Unique, LEGAL VALUES, CODES



Page Name: [Blank]

City, State, Country, County, District, Block, House, Street, Date, Time, Age, Sex, Education, Occupation, Religion, Marital Status, Household Income, Telephone, Fax, Email, Other Data

Do Epidemiologic Analysis

Person

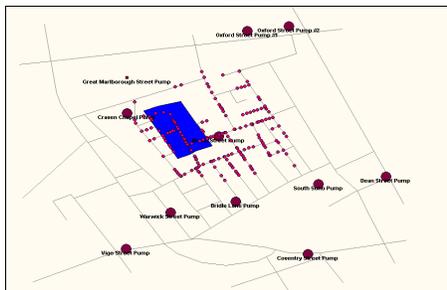
VANILLA (+)	(-)	TOTAL	
(+)	43	11	54
(-)	3	18	21
TOTAL	46	29	75

PARAMETERS	Odds-based	Point Estimate	95% Confidence Interval
			Lower Upper
Odds Ratio (cross product)	23.4545	5.8410	94.1811 (T)
Odds Ratio (MLE)	22.1490	5.9280	109.1500 (M)
		5.2153	138.3900 (R)

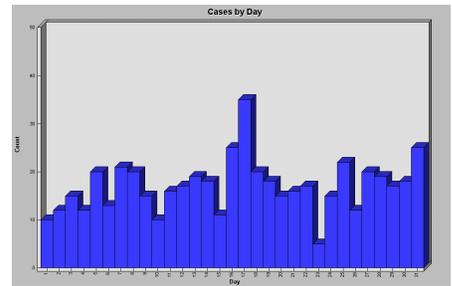
PARAMETERS	Risk-based	Point Estimate	95% Confidence Interval
			Lower Upper
Risk Ratio (RR)	5.5741	1.9283	16.0296 (T)
Risk Difference (RD)	65.3439	46.9212	83.7666 (T)

(T=Taylor series; C=Cochranfield; M=ML; P=Fisher Exact)

Place



Time



Develop a Permanent Data System

Customized Menu



Programming Data Entry

Check Commands

Choose field where action will occur

* DateEntered

Before or After: Before, Alter

User Interaction, Fields, Records, Programs, Variables

Define, Assign

An Analysis Program

```

Program Editor - Statistics
File Edit View Fonts Run Help
New Open Save Print
READ 'Sample.mdb':viewOswego
LIST *
ROUTEOUT Stats.htm
LIST *
FREQ ILL
READ 'Sample.mdb':viewAgeWithCount
FREQ Age WEIGHTVAR=Count
READ 'Sample.mdb':viewEvansCounty
TABLES CAT CHD
TABLES CAT CHD STRATAVAR= AGE1
    
```

Note: The previous page is the back cover. This is the end of the Epi Info Manual

